Networking protocols and administration

ITV8030

lecture 5 part 1:

routing table management, wifi

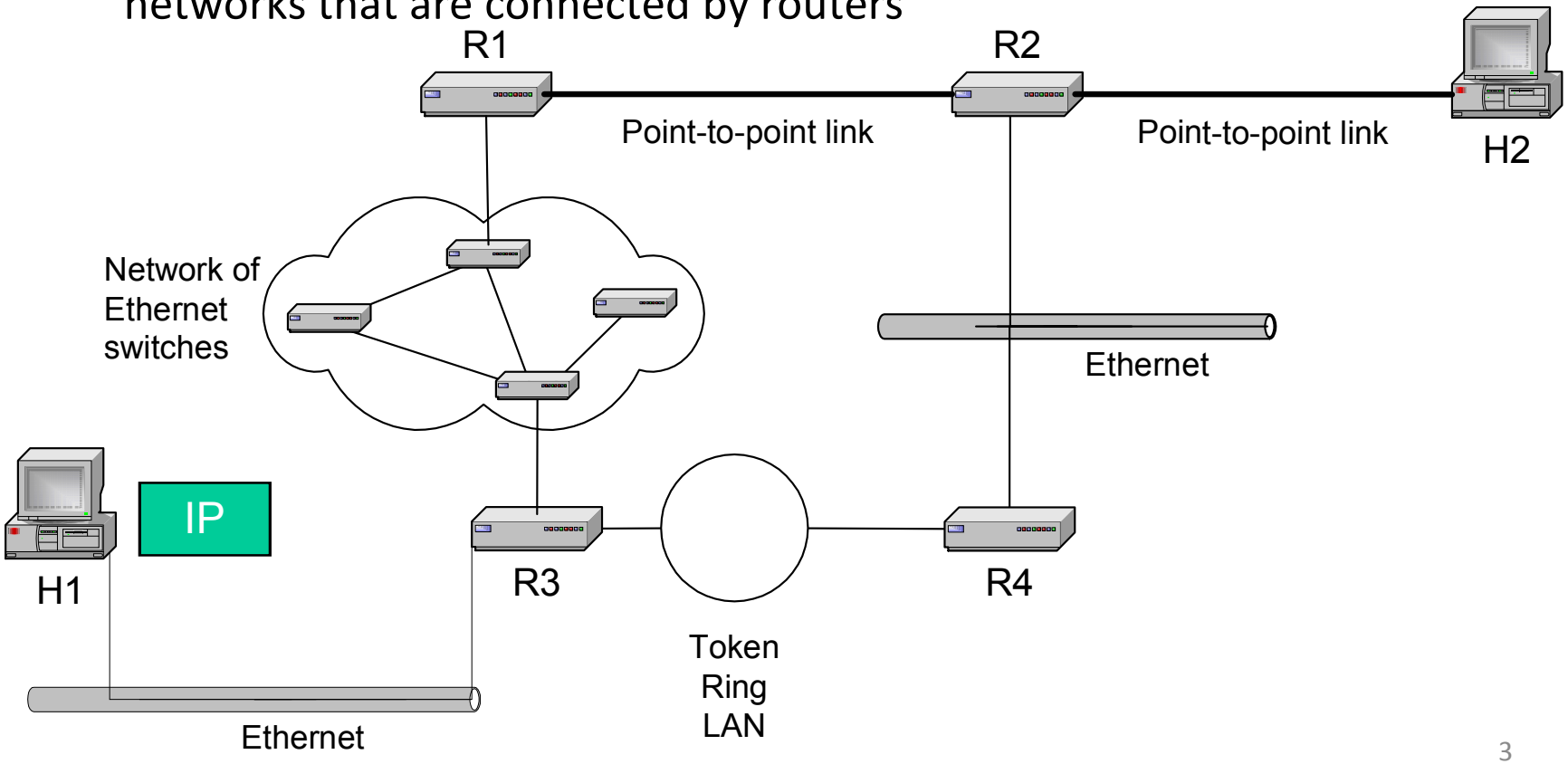
lecture plan

- Refresher: forwarding and routing, ICMP
- Fairly complex stuff, just for basic overview:
  - Routing table management principles
  - Routing table management protocols:
    - RIP
    - OSPF
- WIFI

- Using mostly slides from Univ of Virginia / Univ of Toronto

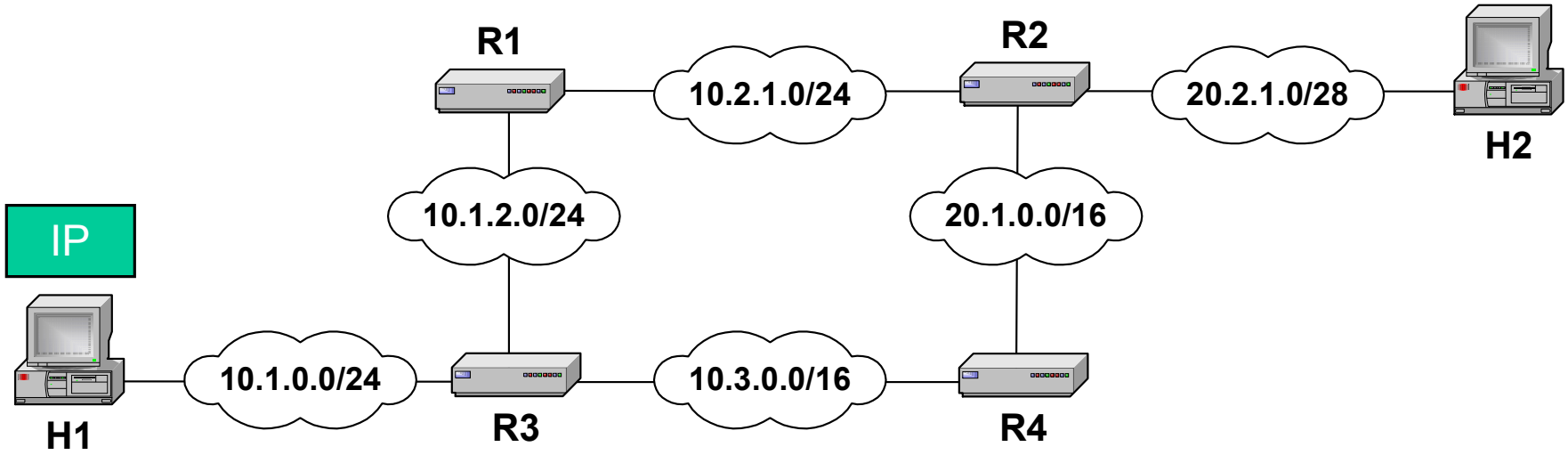
Delivery of an IP datagram

- View at the data link layer layer:
  - Internetwork is a collection of LANs or point-to-point links or switched networks that are connected by routers



Delivery of an IP datagram

- View at the IP layer:
  - An IP network is a logical entity with a network number
  - We represent an IP network as a “cloud”
  - The IP delivery service takes the view of clouds, and ignores the data link layer view



Routing tables

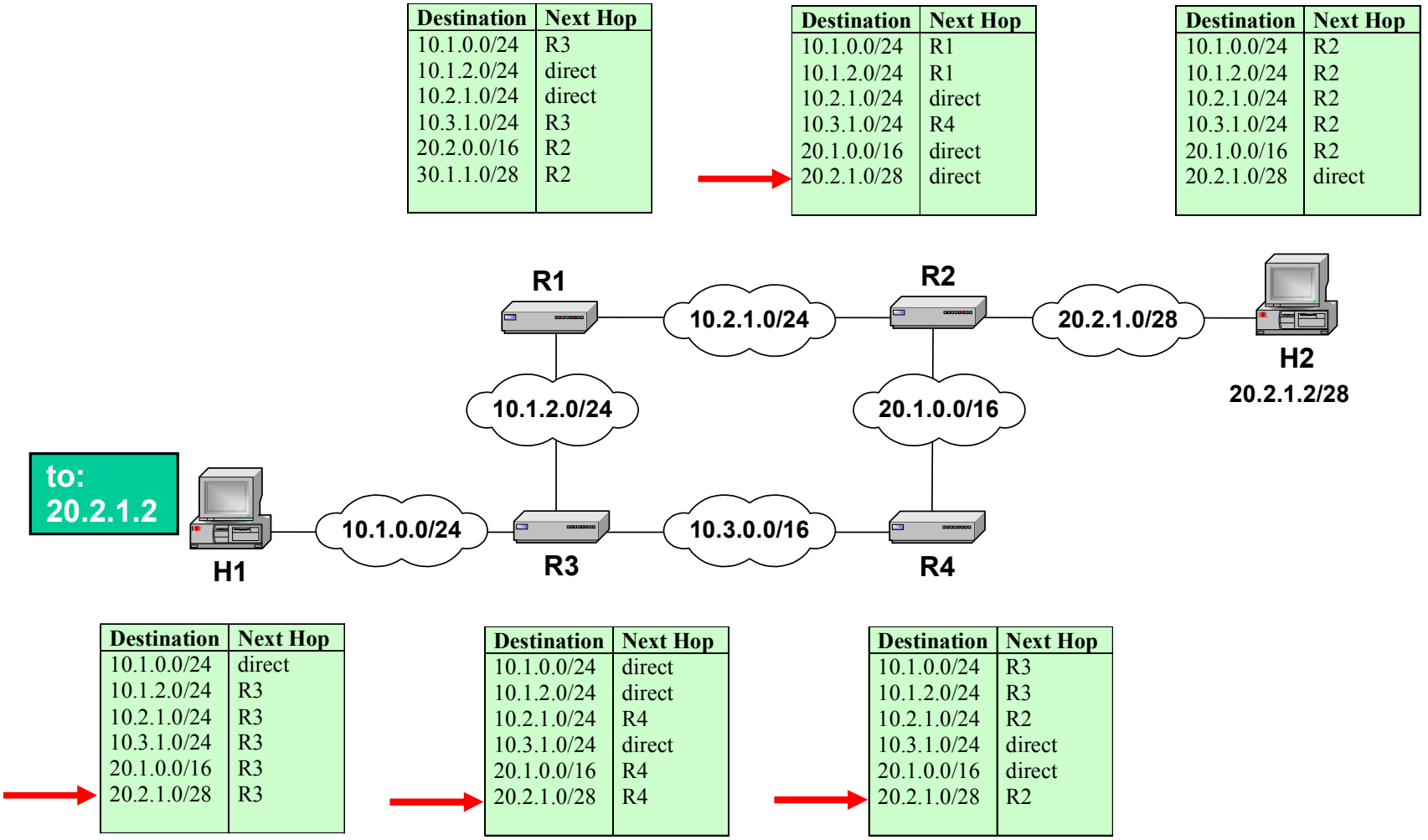
- Each router and each host keeps a **routing table** which tells the router how to process an outgoing packet
- Main columns:
  1. **Destination address:** where is the IP datagram going to?
  2. **Next hop:** how to send the IP datagram?
  3. **Interface:** what is the output port?
- Next hop and interface column can often be summarized as one column
- Routing tables are set so that datagrams gets closer to the its destination

Routing table of a host or router

IP datagrams can be directly delivered ("direct") or is sent to a router ("R4")

Destination	Next Hop	interface
10.1.0.0/24	direct	eth0
10.1.2.0/24	direct	eth0
10.2.1.0/24	R4	serial0
10.3.1.0/24	direct	eth1
20.1.0.0/16	R4	eth0
20.2.1.0/28	R4	eth0

Delivery with routing tables



Delivery of IP datagrams

- There are two distinct processes to delivering IP datagrams:
  1. **Forwarding:** How to pass a packet from an input interface to the output interface?
  2. **Routing:** How to find and setup the routing tables?

- Forwarding must be done as fast as possible:
  - on routers, is often done with support of hardware
  - on PCs, is done in kernel of the operating system
- Routing is less time-critical
  - On a PC, routing is done as a background process

Processing of an IP datagram at a router

1. IP header validation
2. Process options in IP header
3. Parsing the destination IP address
4. Routing table lookup
5. Decrement TTL
6. Perform fragmentation (if necessary)
7. Calculate checksum
8. Transmit to next hop
9. Send ICMP packet (if necessary)

Routing table lookup

- When a router or host need to transmit an IP datagram, it performs a routing table lookup
- **Routing table lookup:** Use the IP destination address as a key to search the routing table.
- Result of the lookup is the IP address of a next hop router, and/or the name of a network interface

Destination address	**Next hop/ interface**
network prefix *or* host IP address *or* loopback address *or* default route	IP address of next hop router *or* Name of a network interface

Type of routing table entries

- **Network route**
  - Destination addresses is a network address (e.g., 10.0.2.0/24)
  - Most entries are network routes
- **Host route**
  - Destination address is an interface address (e.g., 10.0.1.2/32)
  - Used to specify a separate route for certain hosts
- **Default route**
  - Used when no network or host route matches
  - The router that is listed as the next hop of the default route is the **default gateway (for Cisco: "gateway of last resort)**
- **Loopback address**
  - Routing table for the loopback address (127.0.0.1)
  - The next hop lists the loopback (lo0) interface as outgoing interface

Routing table lookup: Longest Prefix Match

- **Longest Prefix Match:** Search for the routing table entry that has the longest match with the prefix of the destination IP address

1. Search for a match on all 32 bits
2. Search for a match for 31 bits

   …..

32. Search for a mach on 0 bits

Host route, loopback entry
→ 32-bit prefix match

Default route is represented as 0.0.0.0/0
→ 0-bit prefix match

128.143.71.21

Destination address	Next hop
10.0.0.0/8	R1
128.143.0.0/16	R2
128.143.64.0/20	R3
128.143.192.0/20	R3
128.143.71.55/32	R3
default	R5

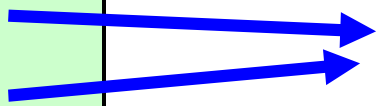
The longest prefix match for 128.143.71.21 is for 24 bits with entry 128.143.71.0/24

Datagram will be sent to R4

Route Aggregation

- Longest prefix match algorithm permits to aggregate prefixes with identical next hop address to a single entry
- This contributes significantly to reducing the size of routing tables of Internet routers

Destination	Next Hop
10.1.0.0/24	R3
10.1.2.0/24	direct
10.2.1.0/24	direct
10.3.1.0/24	R3
20.2.0.0/16	R2
30.1.1.0/28	R2



Destination	Next Hop
10.1.0.0/24	R3
10.1.2.0/24	direct
10.2.1.0/24	direct
10.3.1.0/24	R3
20.0.0.0/8	R2

How do routing tables get updated?

- Adding an interface:
  - Configuring an interface eth2 with 10.0.2.3/24 adds a routing table entry:

Destination	Next Hop/ interface
10.0.2.0/24	eth2

- Adding a default gateway:
  - Configuring 10.0.2.1 as the default gateway adds the entry:

Destination	Next Hop/ interface
0.0.0.0/0	10.0.2.1

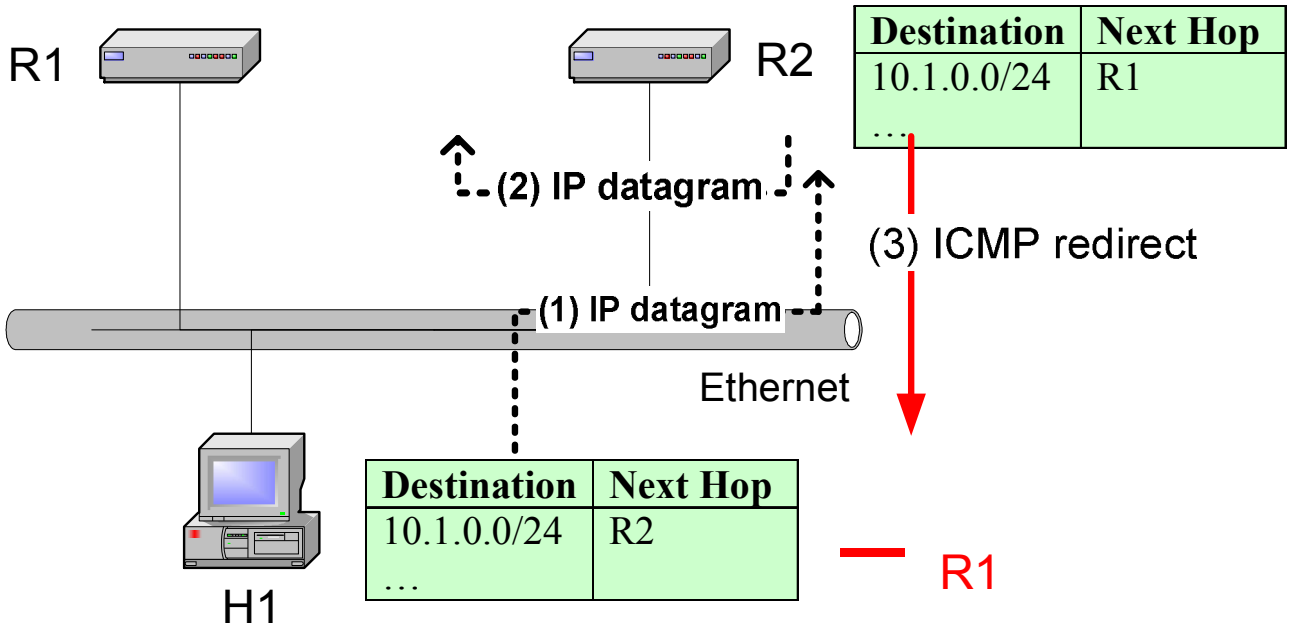
- Static configuration of network routes or host routes
- Update of routing tables through routing protocols
- ICMP messages

Updating the routing table

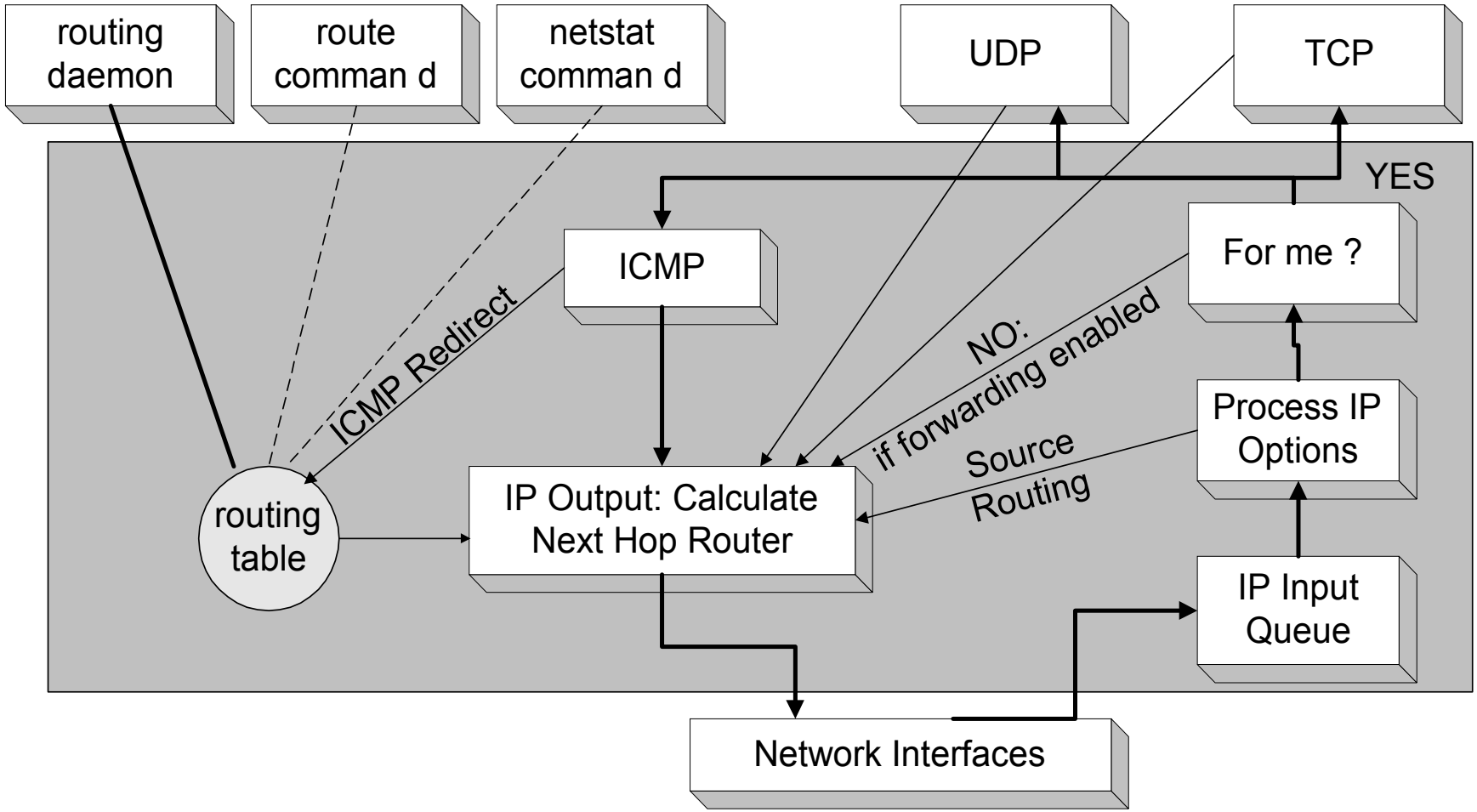
- First possibility: **static routing** (manual updating)
  - fully manual or
  - additionally using ICMP (error messages)
- Third possibility: **dynamic routing** (special routing table protocols for automatic routing table forwarding and information sharing between routers):
  - Intradomain: **RIP, OSPF**
  - Interdomain: EGP, BGP

Routing table manipulations with ICMP

- When a router detects that an IP datagram should have gone to a different router, the router (here R2)
  - forwards the IP datagram to the correct router
  - sends an ICMP redirect message to the host
- Host uses ICMP message to update its routing table



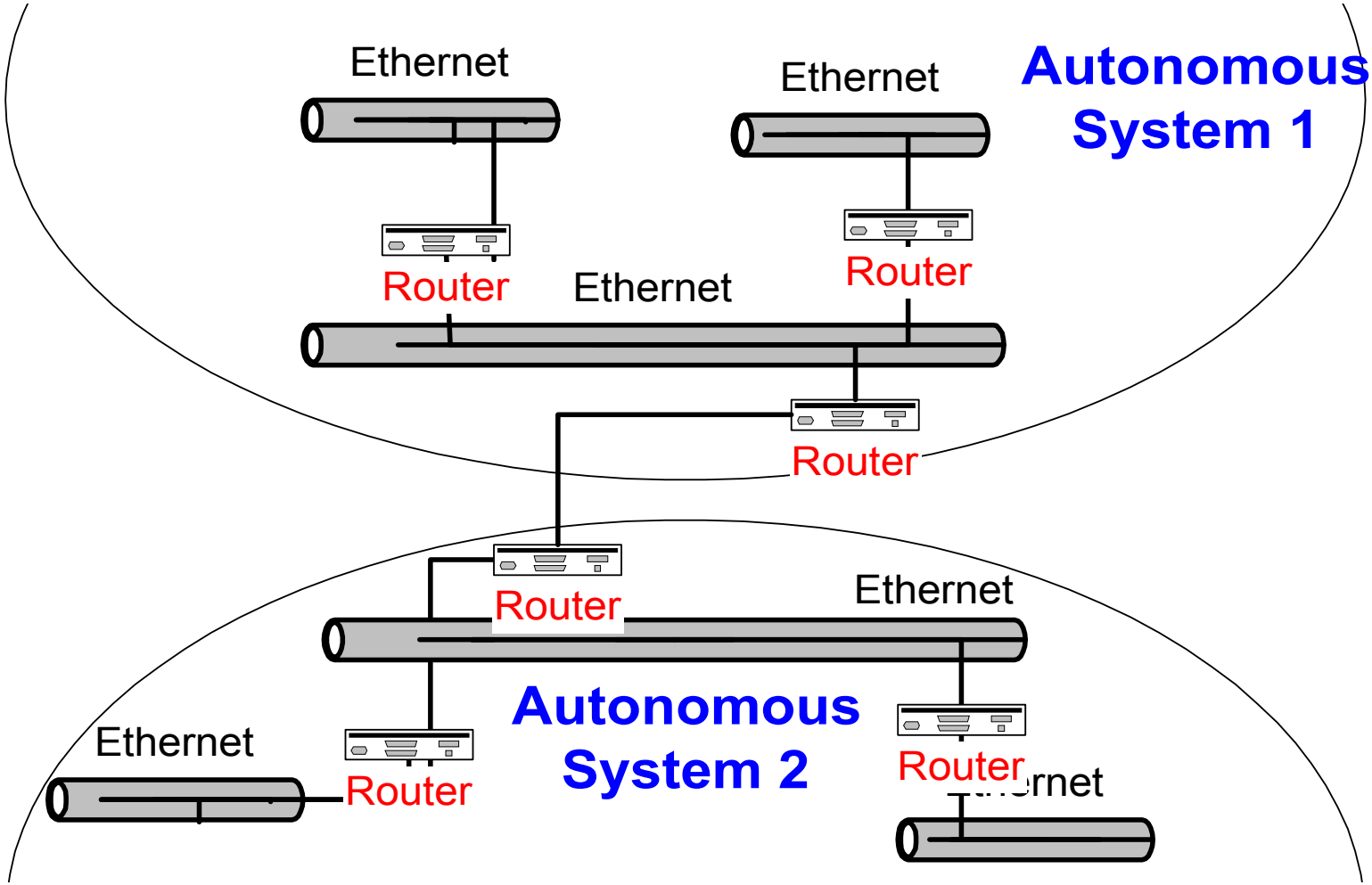
IP Routing


routing daemon
route comman d
netstat comman d
UDP
TCP
YES
ICMP
For me ?
ICMP Redirect
NO:
if forwarding enabled
Process IP Options
Source Routing
routing table
IP Output: Calculate Next Hop Router
IP Input Queue
Network Interfaces

Autonomous Systems

- An **autonomous system** is a region of the Internet that is administered by a single entity.
- Examples of autonomous regions are:
  - UVA's campus network
  - MCI's backbone network
  - Regional Internet Service Provider
- Routing is done differently within an autonomous system (**intradomain routing**) and between autonomous system **(interdomain routing**).

Autonomous Systems (AS)



Interdomain and Intradomain Routing

Intradomain Routing

- Routing within an AS
- Ignores the Internet outside the AS
- Protocols for Intradomain routing are also called **Interior Gateway Protocols** or **IGP's**.
- Popular protocols are
  - RIP (simple, old)
  - OSPF (better)

Interdomain Routing

- Routing between AS's
- Assumes that the Internet consists of a collection of interconnected AS's
- Normally, there is one dedicated router in each AS that handles interdomain traffic.
- Protocols for interdomain routing are also called **Exterior Gateway Protocols** or **EGP's.**
- Routing protocols:
  - EGP
  - BGP (more recent)

Components of a Routing Algorithm

- A procedure for sending and receiving reachability information about network to other routers

- A procedure for calculating optimal routes
  - Routes are calculated using a shortest path algorithm:
    - **Goal**: Given a network were each link is assigned a cost. Find the path with the least cost between two networks with minimum cost.

- A procedures for reacting to and advertising

Approaches to Shortest Path Routing

- There are two basic routing algorithms found on the Internet.

1. Distance Vector Routing

- Each node knows the distance (=cost) to its directly connected neighbors
- A node sends periodically a list of routing updates to its neighbors.
- If all nodes update their distances, the routing tables eventually converge
- New nodes advertise themselves to their neighbors

2. Link State Routing

- Each node knows the distance to its neighbors
- The distance information (=link state) is broadcast to all nodes in the network
- Each node calculates the routing tables independently

Routing Algorithms in the Internet

Distance Vector

- **Routing Information Protocol (RIP)**
- Gateway-to-Gateway Protocol (GGP)
- Exterior Gateway Protocol (EGP)
- Interior Gateway Routing Protocol (IGRP)

Link State

- Intermediate System - Intermediate System (IS-IS)
- **Open Shortest Path First (OSPF)**

Dynamic IP Routing Protocols

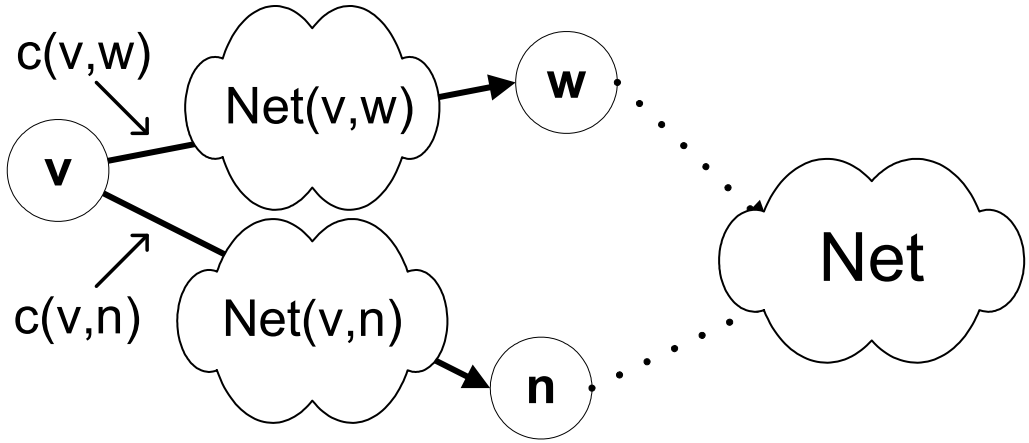
- In Unix systems, the dynamic setting of routing tables is done by the **routed** or **gated** daemons
- The routing daemons execute the following intradomain and interdomain routing protocols

Daemon	*Hello*	*RIP*	*OSPF*	*EGP*	*BGP*
routed		V1			
Gated (Version 3)	Yes	V1 V2	V2	Yes	V2, V3

A network as a graph

- **In the following, networks are represented as a network graph**:
  - nodes are connected by networks
    - network can be a link or a LAN
  - network interface has cost
  - networks are destinations
  - Net(v,w) is an IP address of a network

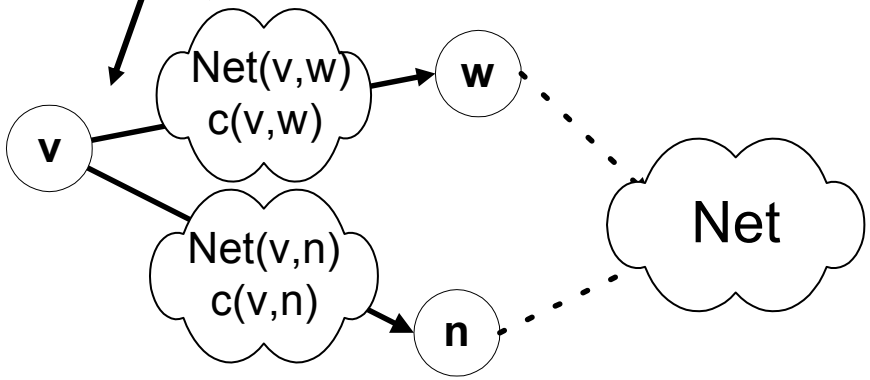
- For ease of notation, we often replace the clouds between nodes by simple links.



Distance Vector Algorithm: Routing Table

c(v,w): cost to transmit on the interface to network Net(v,w)

Net(v,w): Network address of the network between v and w
The network can be a link, but could also be a LAN



RoutingTable of node v

Dest	via (next hop)	cost
Net	n	D(v,Net)

Distance Vector Algorithm: Messages

RoutingTable of node v

Dest	via (next hop)	cost
Net	n	D(v,Net)

Nodes send messages to their neighbors which contain routing table entries

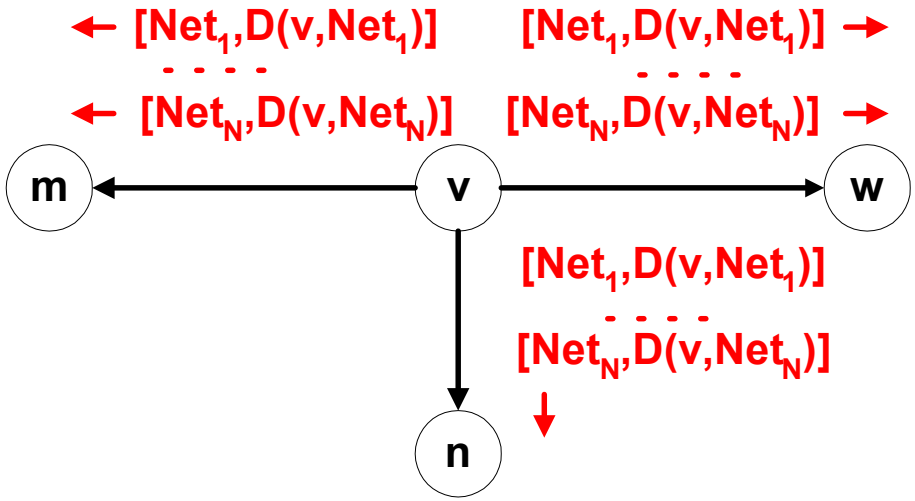
A message has the format: **[Net , D(v,Net)]** means**"My cost to go to Net is D (v,Net)"**

Distance Vector Algorithm: Sending Updates

RoutingTable of node v

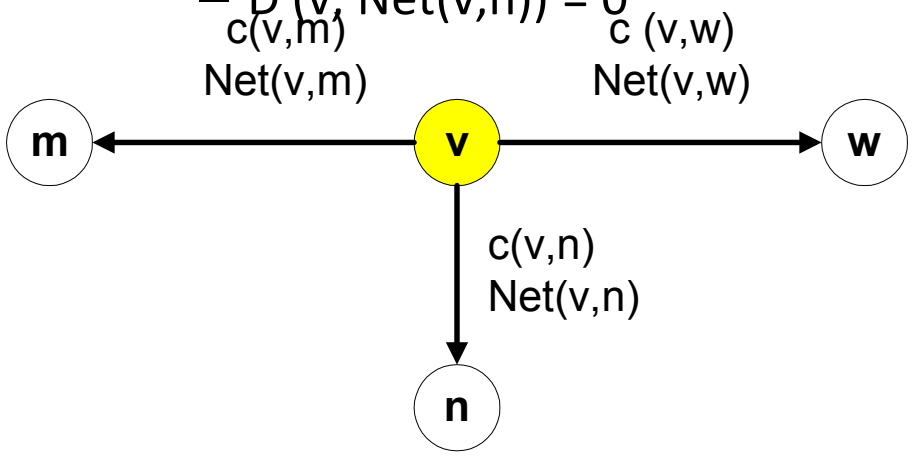
Dest	via (next hop)	cost
Net_1	m	$D(v,Net_1)$
Net_2	n	$D(v,Net_2)$
- - -	- - -	- - -
Net_N	w	$D(v,Net_N)$

Periodically, each node v sends the content of its routing table to its neighbors:



Initiating Routing Table I

- Suppose a new node v becomes active.
- The cost to access directly connected networks is zero:
  - D (v, Net(v,m)) = 0
  - D (v, Net(v,w)) = 0
  - D (v, Net(v,n)) = 0



RoutingTable

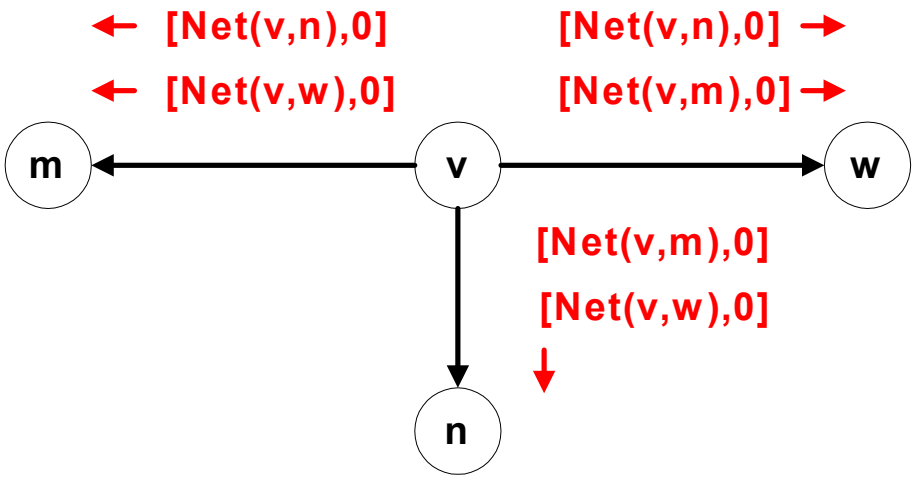
Dest	via (next hop)	cost
Net(v,m)	m	0
Net(v,w)	w	0
Net(v,n)	n	0

Initiating Routing Table II

RoutingTable

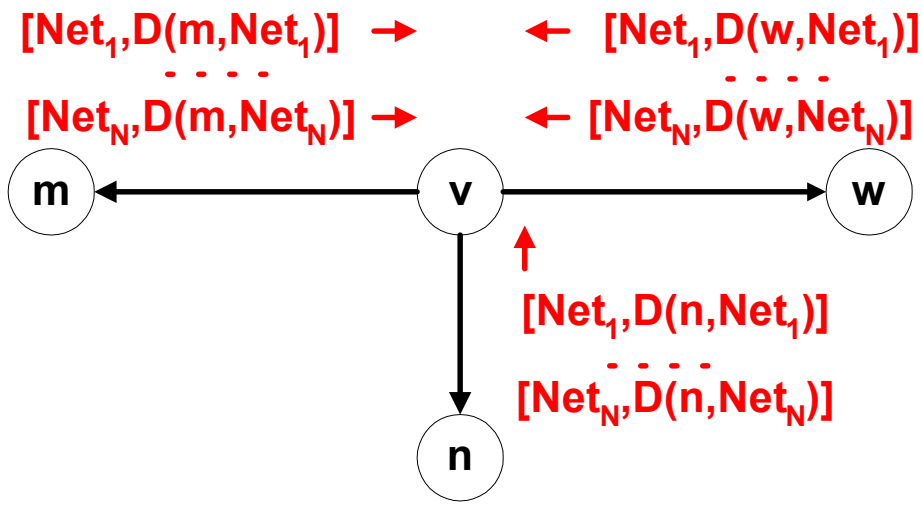
Dest	via (next hop)	cost
Net(v,m)	m	0
Net(v,w)	w	0
Net(v,n)	n	0

- New node v sends the routing table entry to all its



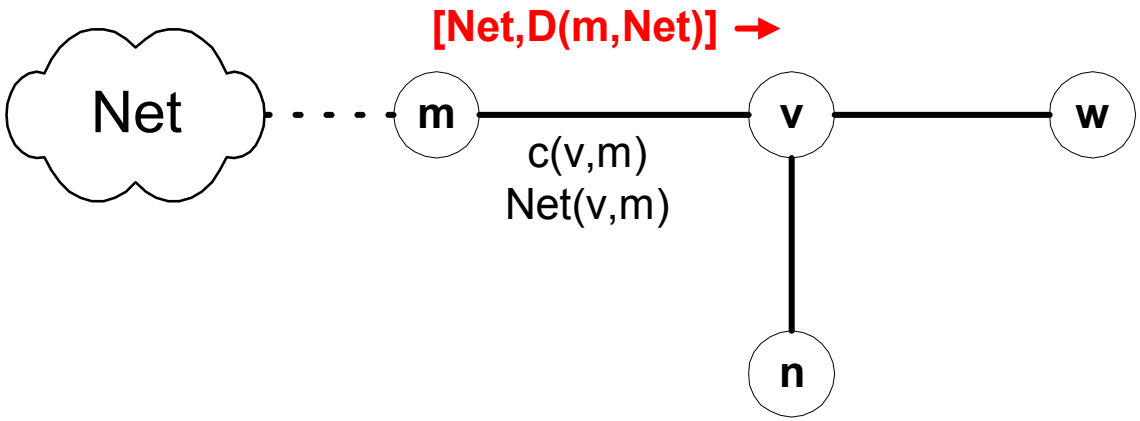
Initiating Routing Table III

- Node v receives the routing tables from other nodes and builds up its routing table



Updating Routing Tables I

Suppose node v receives a message from node m: **[Net,D(m,Net)]**

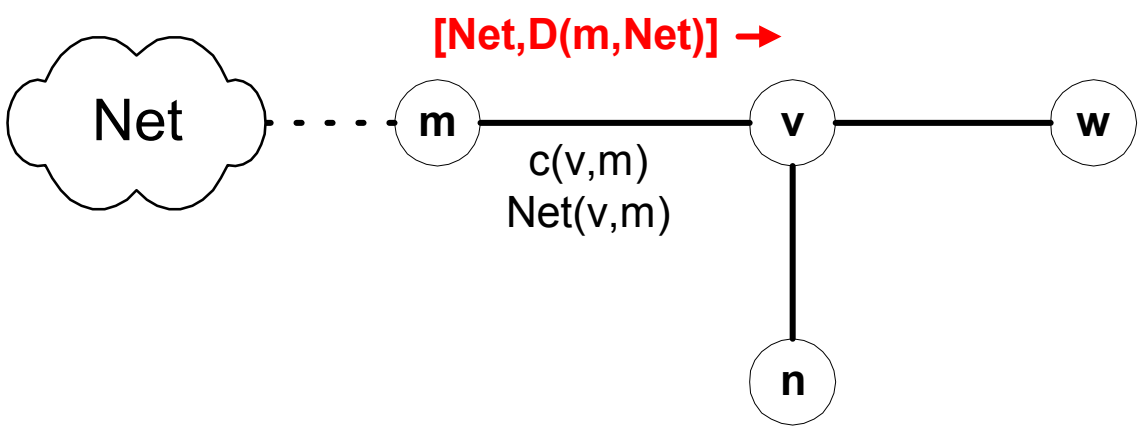


Node v updates its routing table and sends out further messages if the message reduces the cost of a route:

```
if  ( D(m,Net) + c (v,m) < D (v,Net) ) {
      D^new (v,Net) := D (m,Net) + c (v,m);
      Update routing table;
      send message [Net, D^new (v,Net)] to all neighbors
}
```

Updating Routing Tables II

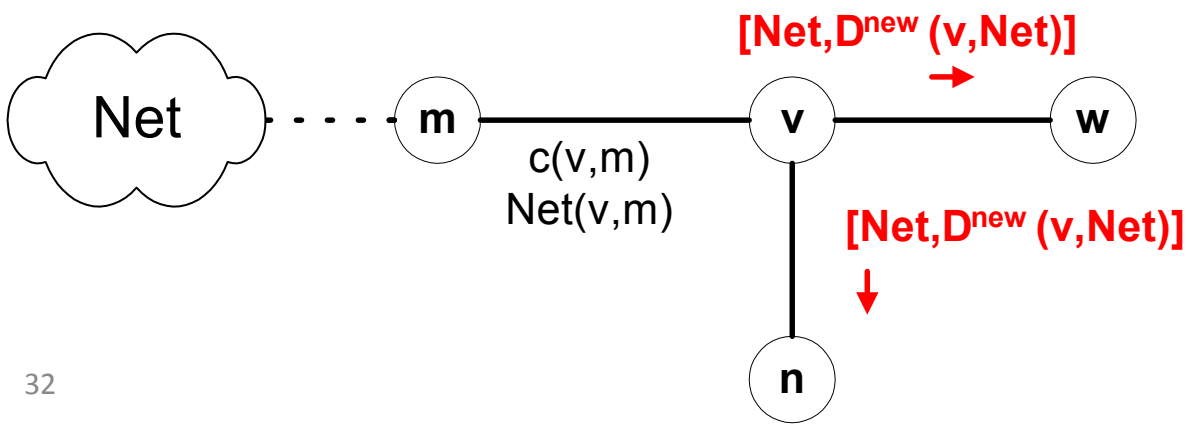
Before receiving the message:



RoutingTable

Dest	via (next hop)	cost
Net	??	D(v,Net)

Suppose D (m,Net) + c (v,m) < D (v,Net):

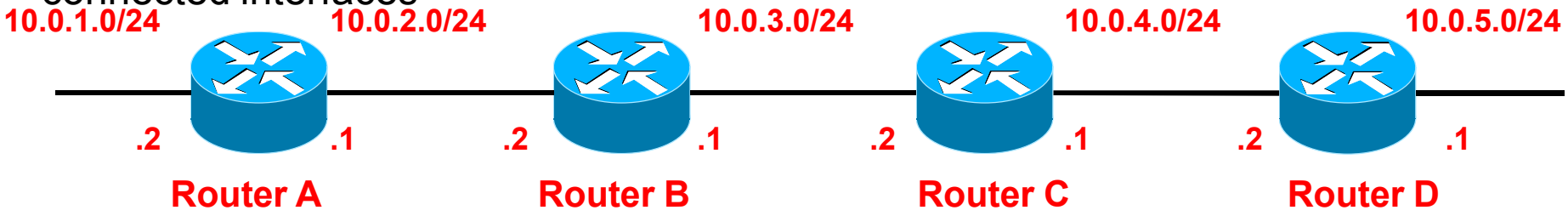


RoutingTable

Dest	via (next hop)	cost
Net	m	$D^{new}(v,Net)$

Example

Assume: - link cost is 1, i.e., c(v,w) = 1
- all updates, updates occur simultaneously
- Initially, each router only knows the cost of connected interfaces



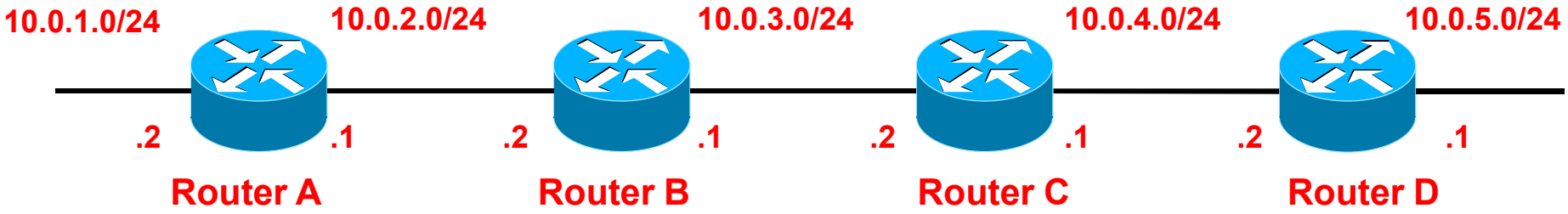
Net	via	cost
t=0:		
10.0.1.0 -	0	
10.0.2.0 -	0	
t=1:		
10.0.1.0 -	0	
10.0.2.0 -	0 10.0.3.0	
10.0.2.2 1		
t=2:		
10.0.1.0 -	0	
10.0.2.0 -	0 10.0.3.0	
10.0.2.2 1		
10.0.4.0 10.0.2.2 2		

Net	via	cost
t=0:		
10.0.2.0 -	0	
10.0.3.0 -	0	
t=1:		
10.0.1.0 10.0.2.1 1		
10.0.2.0 -	0	
10.0.3.0 -	0	
10.0.4.0 10.0.3.2 1		
t=2:		
10.0.1.0 10.0.2.1 1		
10.0.2.0 -	0	
10.0.3.0 -	0	
10.0.4.0 10.0.3.2 1		
10.0.5.0 10.0.3.2 2		

Net	via	cost
t=0:		
10.0.3.0 -	0	
10.0.4.0 -	0	
t=1:		
10.0.2.0 10.0.3.1 1		
10.0.3.0 -	0	
10.0.4.0 -	0	
10.0.5.0 10.0.4.2 1		
t=2:		
10.0.1.0 10.0.3.1 2		
10.0.2.0 10.0.3.1 1		
10.0.3.0 -	0	
10.0.4.0 -	0	
10.0.5.0 10.0.4.2 1		

Net	via	cost
t=0:		
10.0.4.0 -	0	
10.0.5.0 -	0	
t=1:		
10.0.3.0 10.0.4.1 1		
10.0.4.0 -	0	
10.0.5.0 -	0	
t=2:		
10.0.2.0 10.0.4.1 2		
10.0.3.0 10.0.4.1 1		
10.0.4.0 -	0	
10.0.5.0 -	0	

Example



Net	via	cost
t=2:		
10.0.1.0 -	0	
10.0.2.0 -	0	10.0.3.0
10.0.2.2	1	
10.0.4.0	10.0.2.2	2
t=3:		
10.0.1.0 -	**0**	
10.0.2.0 -	**0**	**10.0.3.0**
10.0.2.2	**1**	
10.0.4.0	**10.0.2.2**	**2**
10.0.5.0	**10.0.2.2**	**3**

Net	via	cost
t=2:		
10.0.1.0	10.0.2.1	1
10.0.2.0 -	0	
10.0.3.0 -	0	
10.0.4.0	10.0.3.2	1
10.0.5.0	10.0.3.2	2
t=3:		
10.0.1.0	**10.0.2.1**	**1**
10.0.2.0 -	**0**	
10.0.3.0 -	**0**	
10.0.4.0	**10.0.3.2**	**1**
10.0.5.0	**10.0.3.2**	**2**

Net	via	cost
t=2:		
10.0.1.0	10.0.3.1	2
10.0.2.0	10.0.3.1	1
10.0.3.0 -	0	
10.0.4.0 -	0	
10.0.5.0	10.0.4.2	1
t=3:		
10.0.1.0	**10.0.3.1**	**2**
10.0.2.0	**10.0.3.1**	**1**
10.0.3.0 -	**0**	
10.0.4.0 -	**0**	
10.0.5.0	**10.0.4.2**	**1**

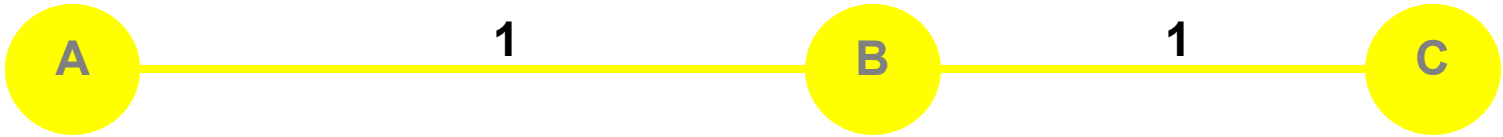
Net	via	cost
t=2:		
10.0.2.0	10.0.4.1	2
10.0.3.0	10.0.4.1	1
10.0.4.0 -	0	
10.0.5.0 -	0	
t=3:		
10.0.1.0	**10.0.4.1**	**3**
10.0.2.0	**10.0.4.1**	**2**
10.0.3.0	**10.0.4.1**	**1**
10.0.4.0 -	**0**	
10.0.5.0 -	**0**	

Now, routing tables have converged !

Characteristics of Distance Vector Routing

- **Periodic Updates:** Updates to the routing tables are sent at the end of a certain time period. A typical value is 90 seconds.
- **Triggered Updates:** If a metric changes on a link, a router immediately sends out an update without waiting for the end of the update period.
- **Full Routing Table Update**: Most distance vector routing protocol send their neighbors the entire routing table (not only entries which change).
- **Route invalidation timers:** Routing table entries are invalid if they are not refreshed. A typical value is to invalidate an entry if no update is received after 3-6 update periods.

The Count-to-Infinity Problem



A's Routing Table

to	via (next hop)	cost
C	B	2

B's Routing Table

to	via (next hop)	cost
C	C	1

now link B-C goes down

C	B	2

C	2

C	oo

C	-	oo

C	oo

C	3

C	A	3

C	B	4

C	4

C	oo

C	-	oo

C	oo

Count-to-Infinity

- The reason for the count-to-infinity problem is that each node only has a “next-hop-view”
- For example, in the first step, A did not realize that its route (with cost 2) to C went through node B

- How can the Count-to-Infinity problem be solved?

Count-to-Infinity

- The reason for the count-to-infinity problem is that each node only has a “next-hop-view”
- For example, in the first step, A did not realize that its route (with cost 2) to C went through node B

- How can the Count-to-Infinity problem be solved?
- **Solution 1:** Always advertise the entire path in an update message (**Path vectors**)
  - If routing tables are large, the routing messages require substantial bandwidth
  - BGP uses this solution

Count-to-Infinity

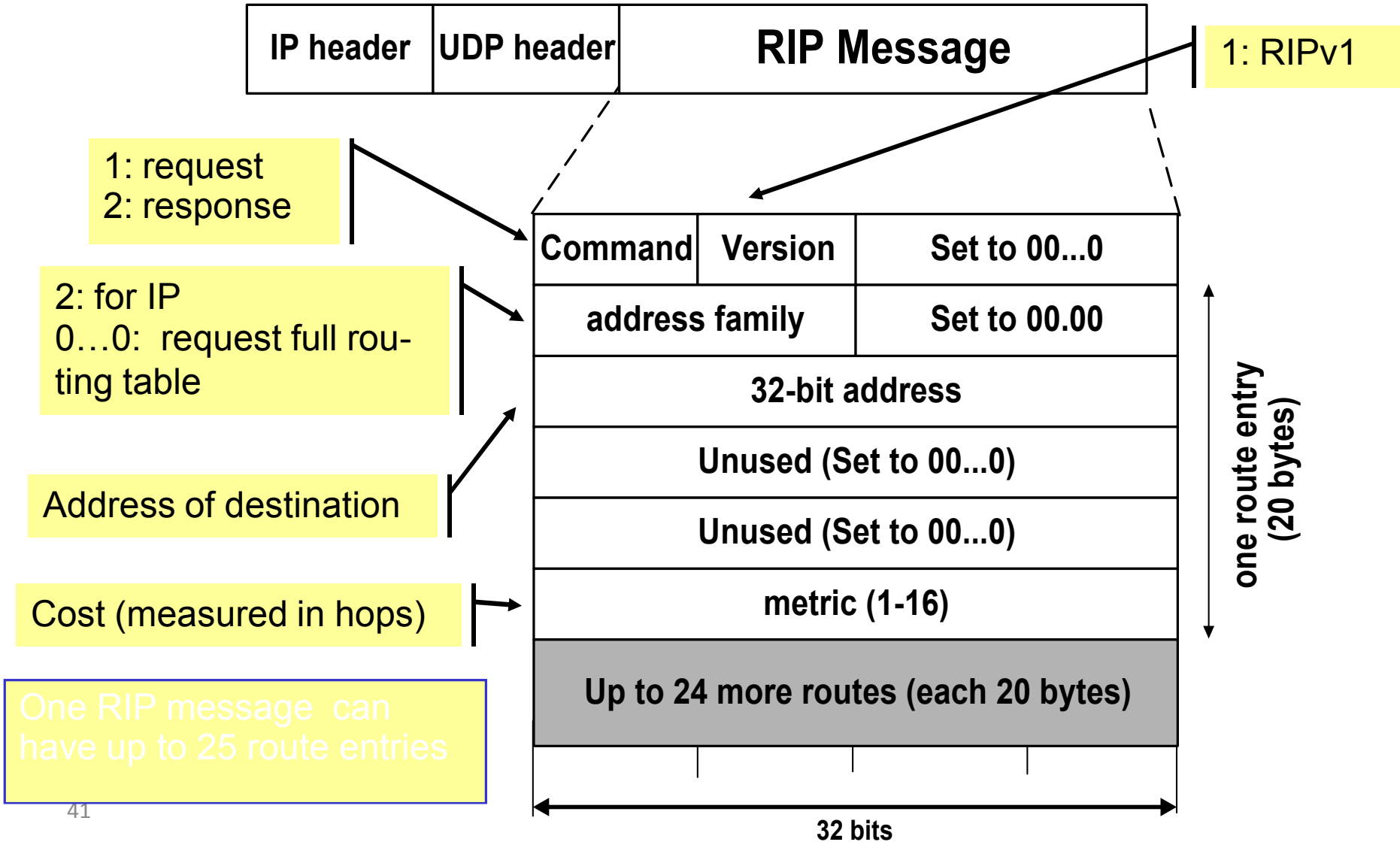
- The reason for the count-to-infinity problem is that each node only has a “next-hop-view”
- For example, in the first step, A did not realize that its route (with cost 2) to C went through node B

- How can the Count-to-Infinity problem be solved?
- **Solution 2:** Never advertise the cost to a neighbor if this neighbor is the next hop on the current path **(Split Horizon)**
  - Example: A would not send the first routing update to B, since B is the next hop on A’s current route to C
  - Split Horizon does not solve count-to-infinity in all cases!

RIP - Routing Information Protocol

- A simple intradomain protocol
- Straightforward implementation of Distance Vector Routing
- Each router advertises its distance vector every 30 seconds (or whenever its routing table changes) to all of its neighbors
- RIP always uses 1 as link metric
- Maximum hop count is 15, with “16” equal to “∞”
- Routes are timeout (set to 16) after 3 minutes if they are not updated

RIPv1 Packet Format

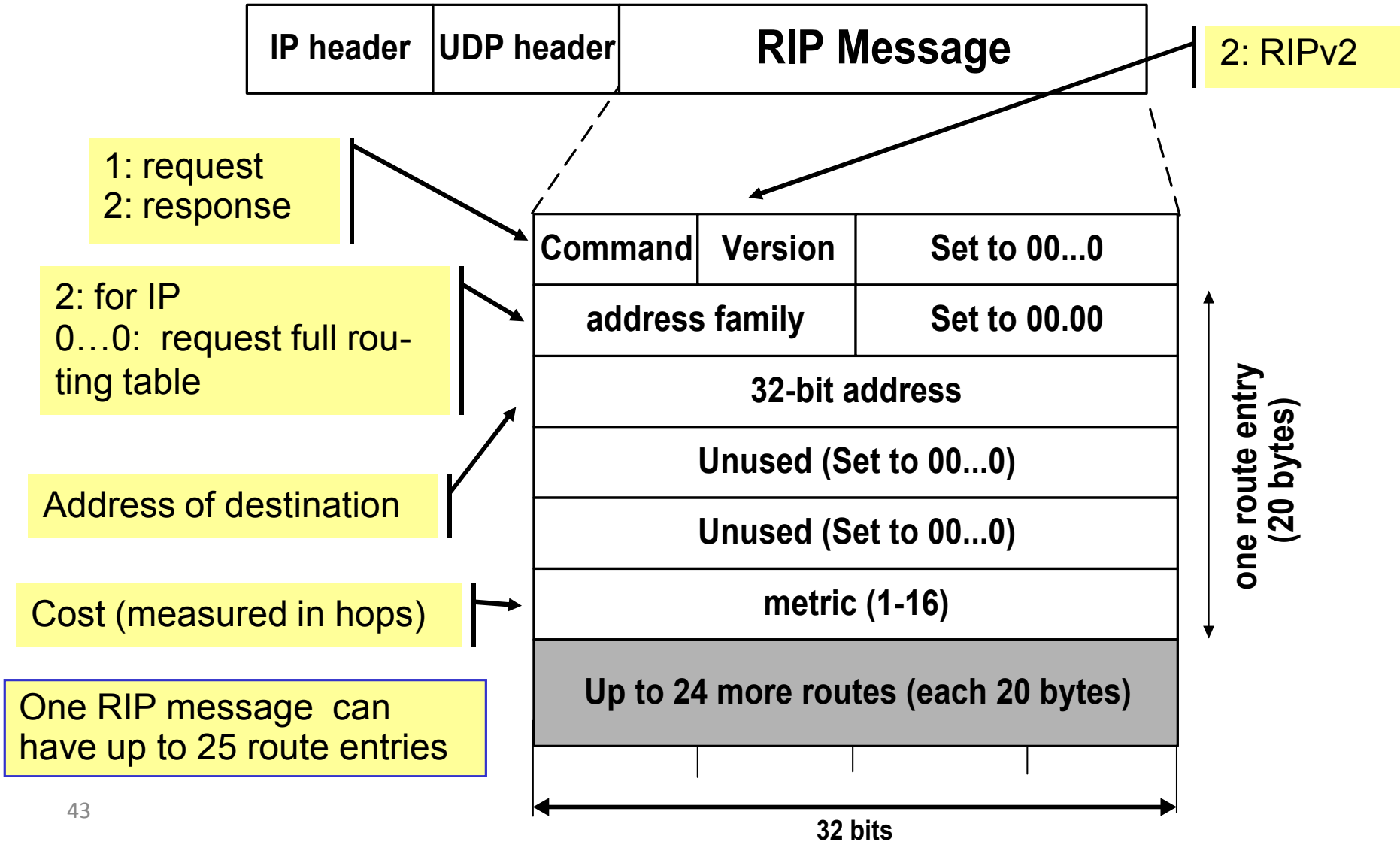


RIPv2

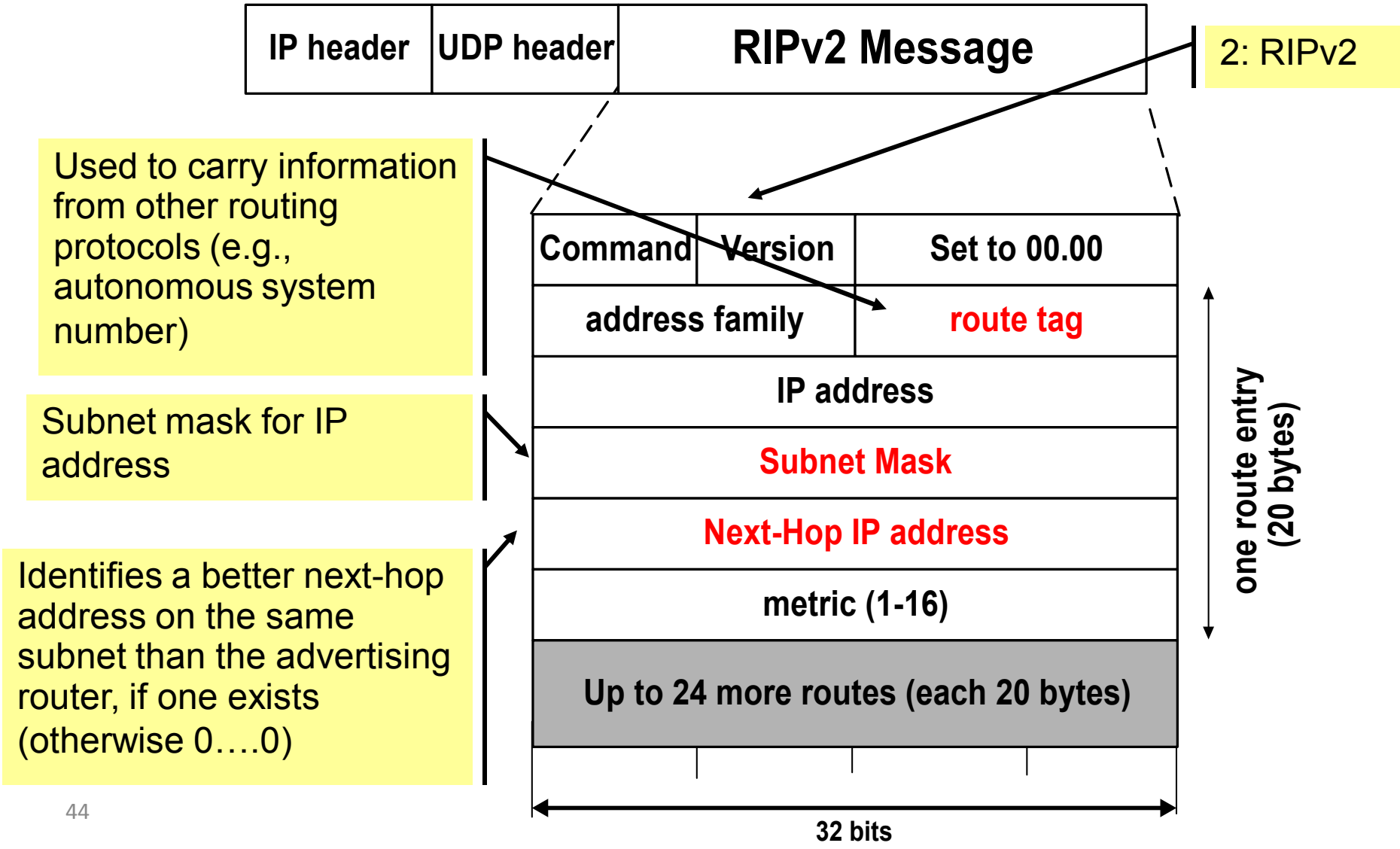
- RIPv2 is an extends RIPv1:
  - Subnet masks are carried in the route information
  - Authentication of routing messages
  - Route information carries next-hop address
  - Exploites IP multicasting

- Extensions of RIPv2 are carried in unused fields of RIPv1 messages

RIPv2 Packet Format



RIPv2 Packet Format



RIP Messages

- This is the operation of RIP in **`routed`**. Dedicated port for RIP is UDP port 520.

- Two types of messages:
  - **Request messages**
    - used to ask neighboring nodes for an update
  - **Response messages**
    - contains an update

Routing with RIP

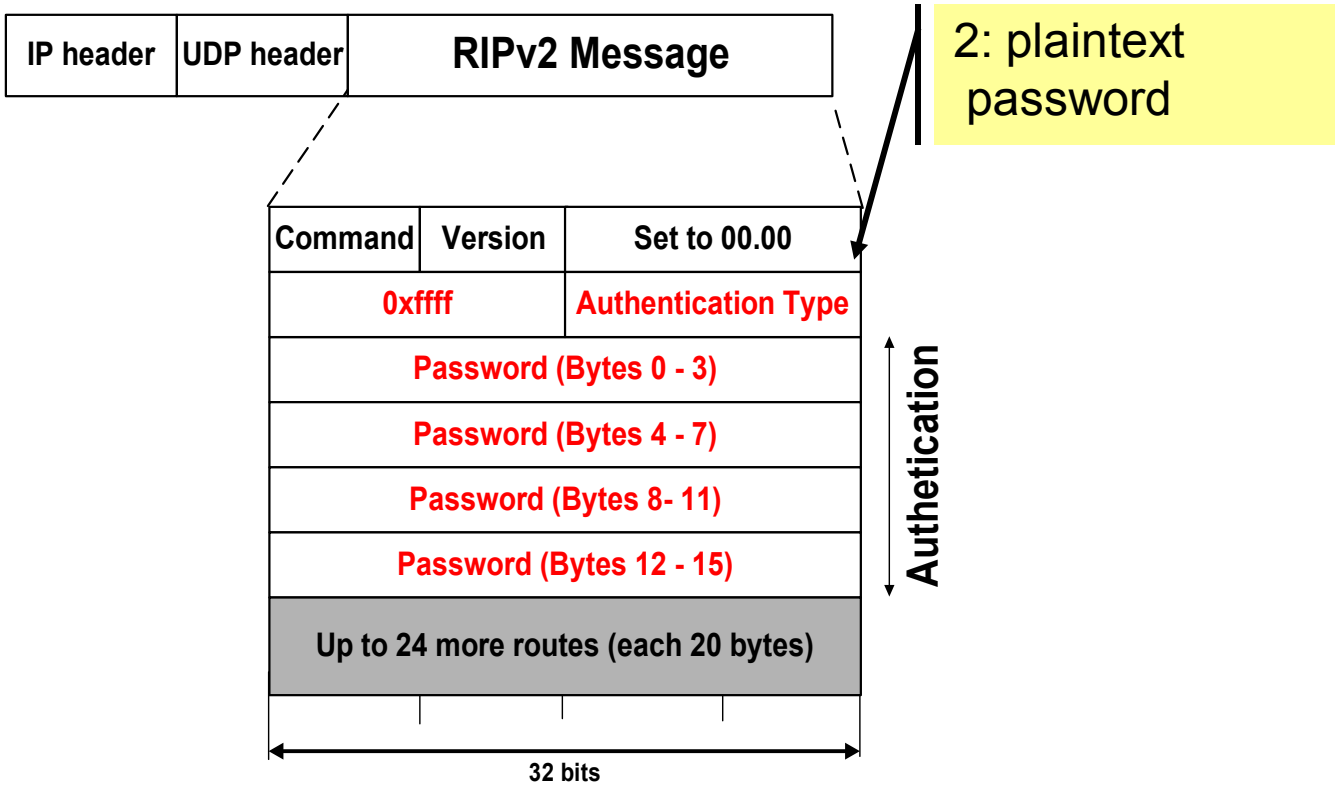
- **Initialization:** Send a **request packet** (command = 1, address family=0..0) on all interfaces:
    - RIPv1 uses broadcast if possible,
    - RIPv2 uses multicast address 224.0.0.9, if possible

  requesting routing tables from neighboring routers
- **Request received**: Routers that receive above request send their entire routing table
- **Response received**: Update the routing table

- **Regular routing updates**: Every 30 seconds, send all or part of the routing tables to every neighbor in an response message
- **Triggered Updates:** Whenever the metric for a route change, send entire routing table.

RIP Security

- Issue: Sending bogus routing updates to a router
- RIPv1: No protection
- RIPv2: Simple authentication scheme

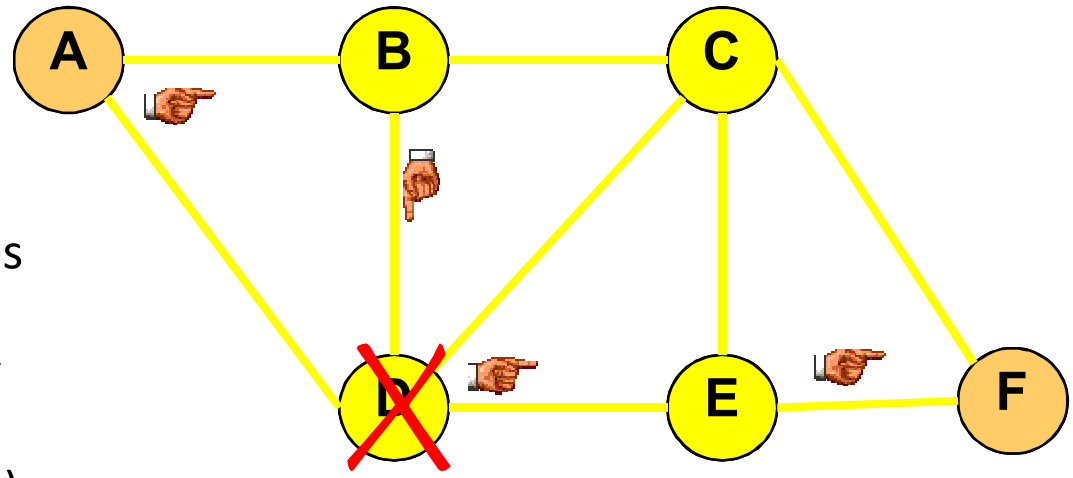


RIP Problems

- RIP takes a long time to stabilize
  - Even for a small network, it takes several minutes until the routing tables have settled after a change
- RIP has all the problems of distance vector algorithms, e.g., count-to-Infinity
    - » RIP uses split horizon to avoid count-to-infinity
- The maximum path in RIP is 15 hops

Distance Vector vs. Link State Routing

- With distance vector routing, each node has information only about the next hop:
  - Node A: to reach F go to B
  - Node B: to reach F go to D
  - Node D: to reach F go to E
  - Node E: go directly to F

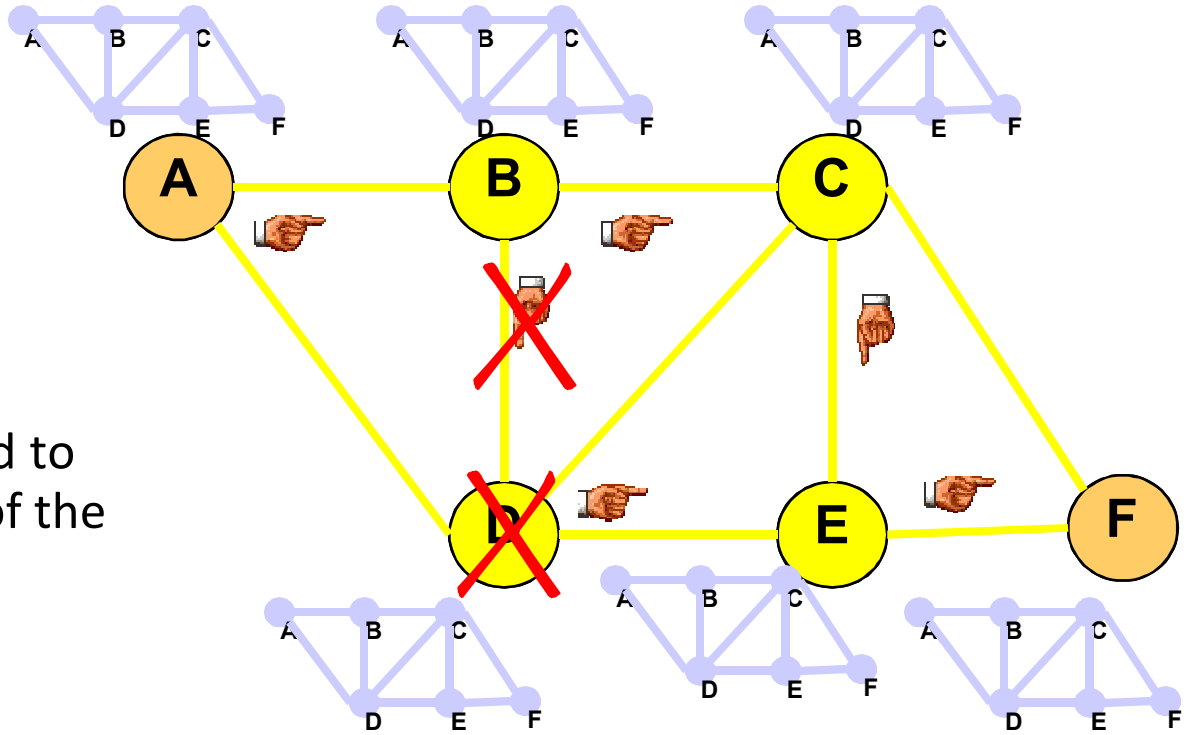


- Distance vector routing makes poor routing decisions if directions are not completely correct
(e.g., because a node is down).

- If parts of the directions incorrect, the routing may be incorrect until the routing algorithms has re-converged.

Distance Vector vs. Link State Routing

- In link state routing, each node has a complete map of the topology
- If a node fails, each node can calculate the new route
- Difficulty: All nodes need to have a consistent view of the network



Link State Routing: Properties

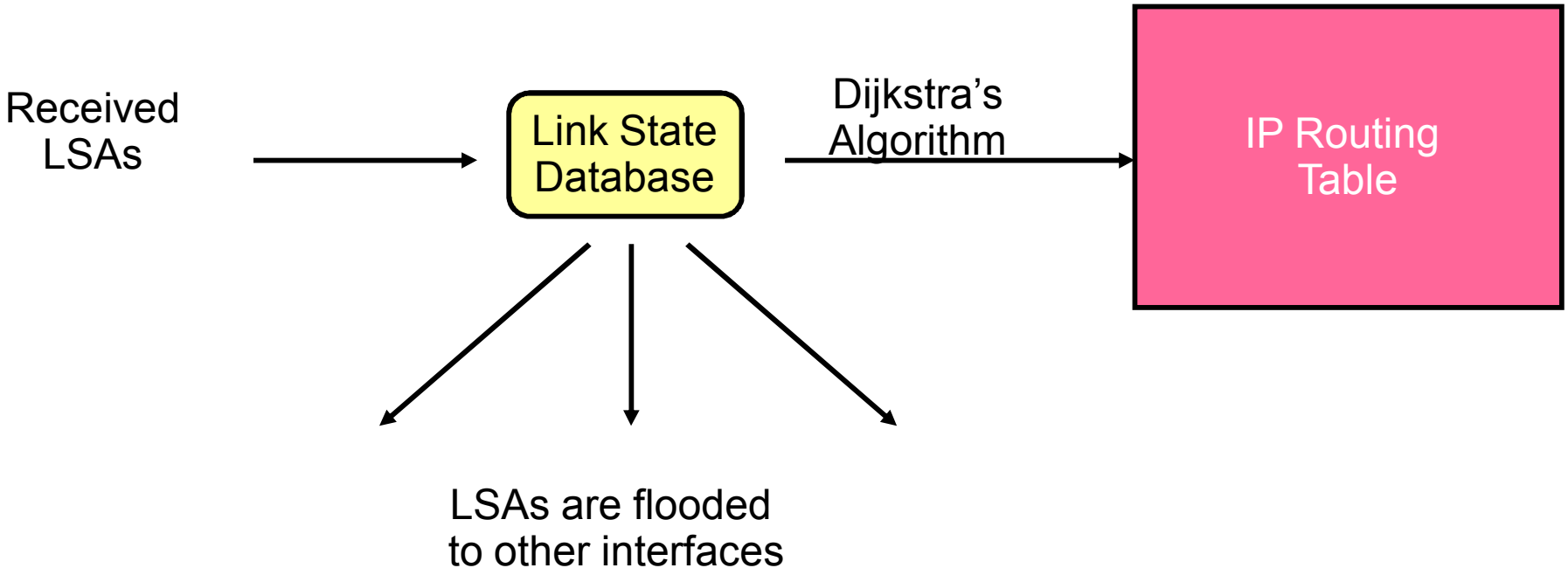
- Each node requires complete topology information
- Link state information must be flooded to all nodes
- Guaranteed to converge

Link State Routing: Basic princples

1. Each router establishes a relationship *("adjacency")* with its neighbors
2. Each router generates *link state advertisements (LSAs)* which are distributed to all routers

   LSA = (link id, state of the link, cost, neighbors of the link)
3. Each router maintains a database of all received LSAs (*topological database* or *link state database*), which describes the network has a graph with weighted edges
4. Each router uses its link state database to run a shortest path algorithm (Dijikstra's algorithm) to produce the shortest path to each network

Operation of a Link State Routing protocol



OSPF

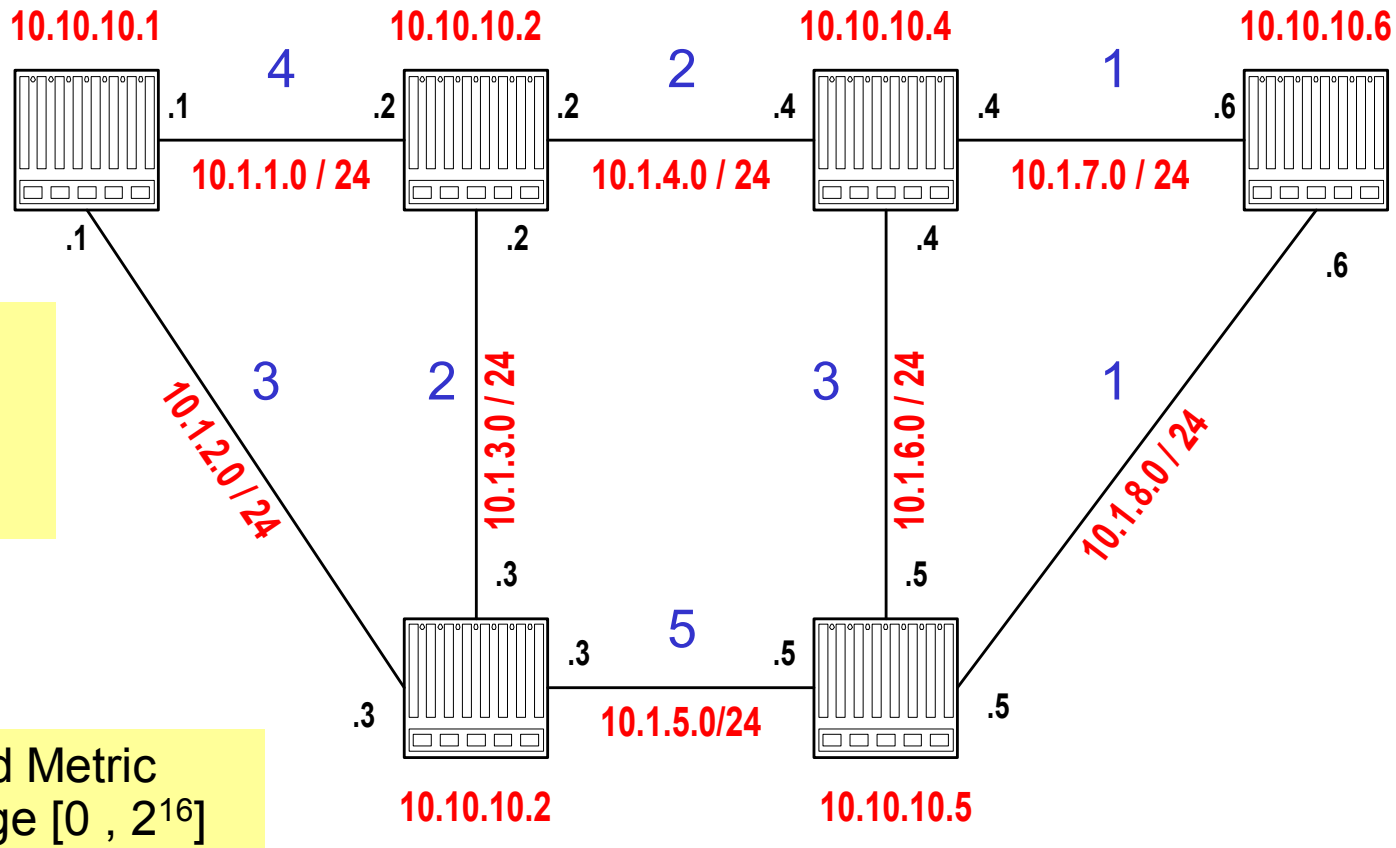
- OSPF = Open Shortest Path First
- The OSPF routing protocol is the most important link state routing protocol on the Internet
- The complexity of OSPF is significant

- History:
  - 1989: RFC 1131 OSPF Version 1
  - 1991: RFC1247 OSPF Version 2
  - 1994: RFC 1583 OSPF Version 2 (revised)
  - 1997: RFC 2178 OSPF Version 2 (revised)
  - 1998: RFC 2328 OSPF Version 2 (current version)

Features of OSPF

- Provides authentication of routing messages
- Enables load balancing by allowing traffic to be split evenly across routes with equal cost
- Type-of-Service routing allows to setup different routes dependent on the TOS field
- Supports subnetting
- Supports multicasting
- Allows hierarchical routing

Example Network

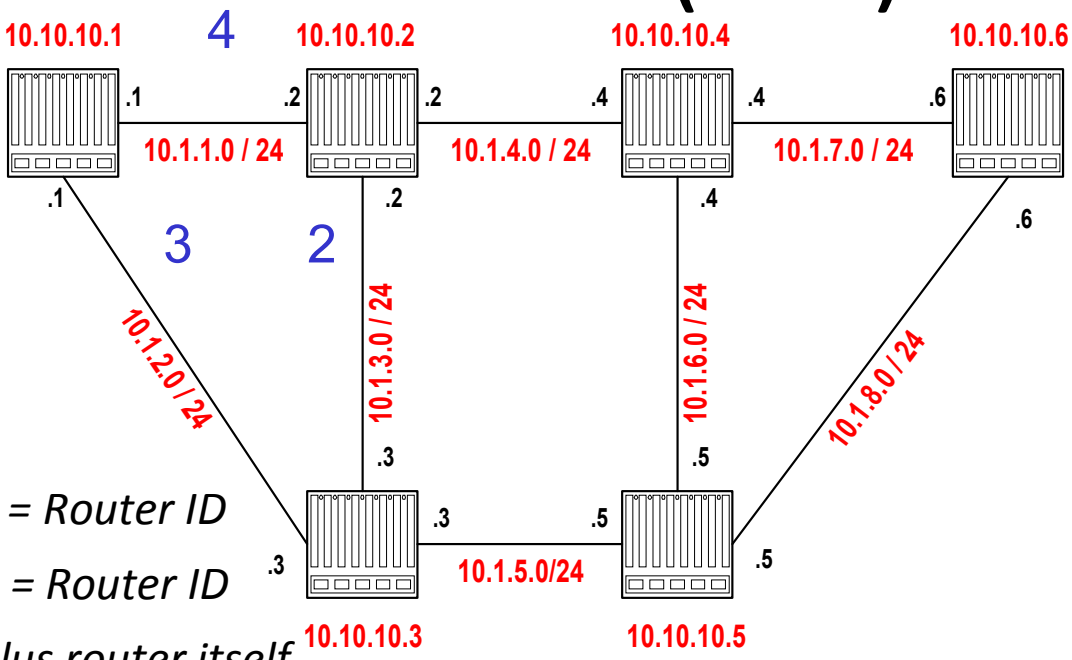


Router IDs are selected independent of interface addresses

Link costs are called Metric
Metric is in the range [0 , 2^{16}]
Metric can be asymmetric

Link State Advertisement (LSA)

- **The LSA of router 10.10.10.1 is as follows:**



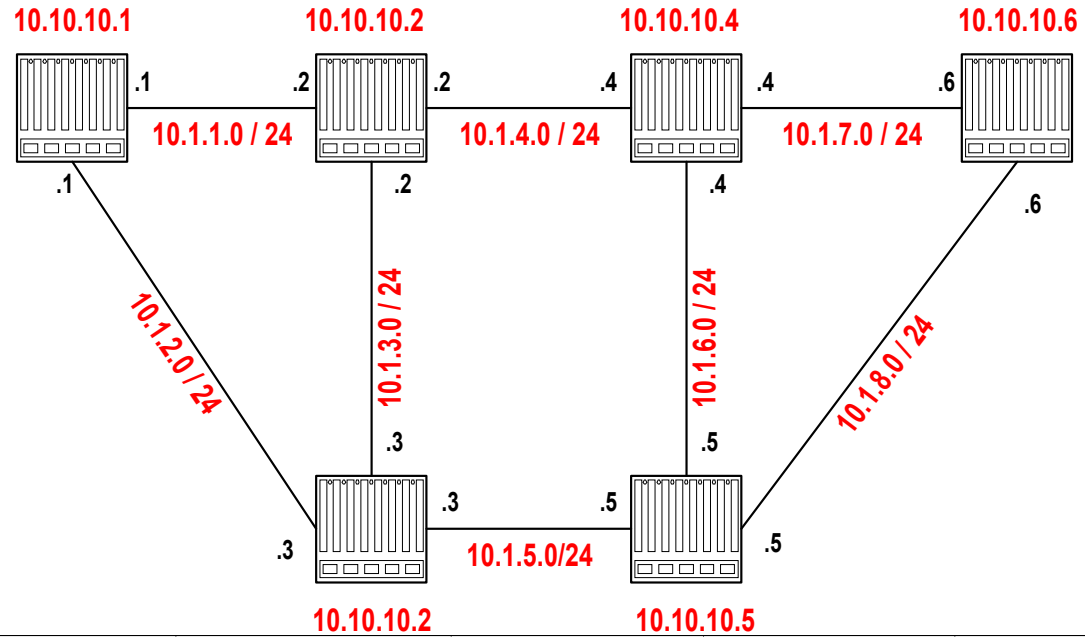
- **Link State ID**: 10.10.10.1 *= Router ID*
- **Advertising Router:** 10.10.10.1 *= Router ID*
- **Number of links:** 3 *= 2 links plus router itself*

- **Description of Link 1:** Link ID = 10.1.1.1, Metric = 4
- **Description of Link 2:** Link ID = 10.1.2.1, Metric = 3
- **Description of Link 3:** Link ID = 10.10.10.1, Metric = 0

Each router sends its LSA to all routers in the network (using a method called reliable flooding)

Network and Link State Database

Each router has a database which contains the LSAs from all other routers

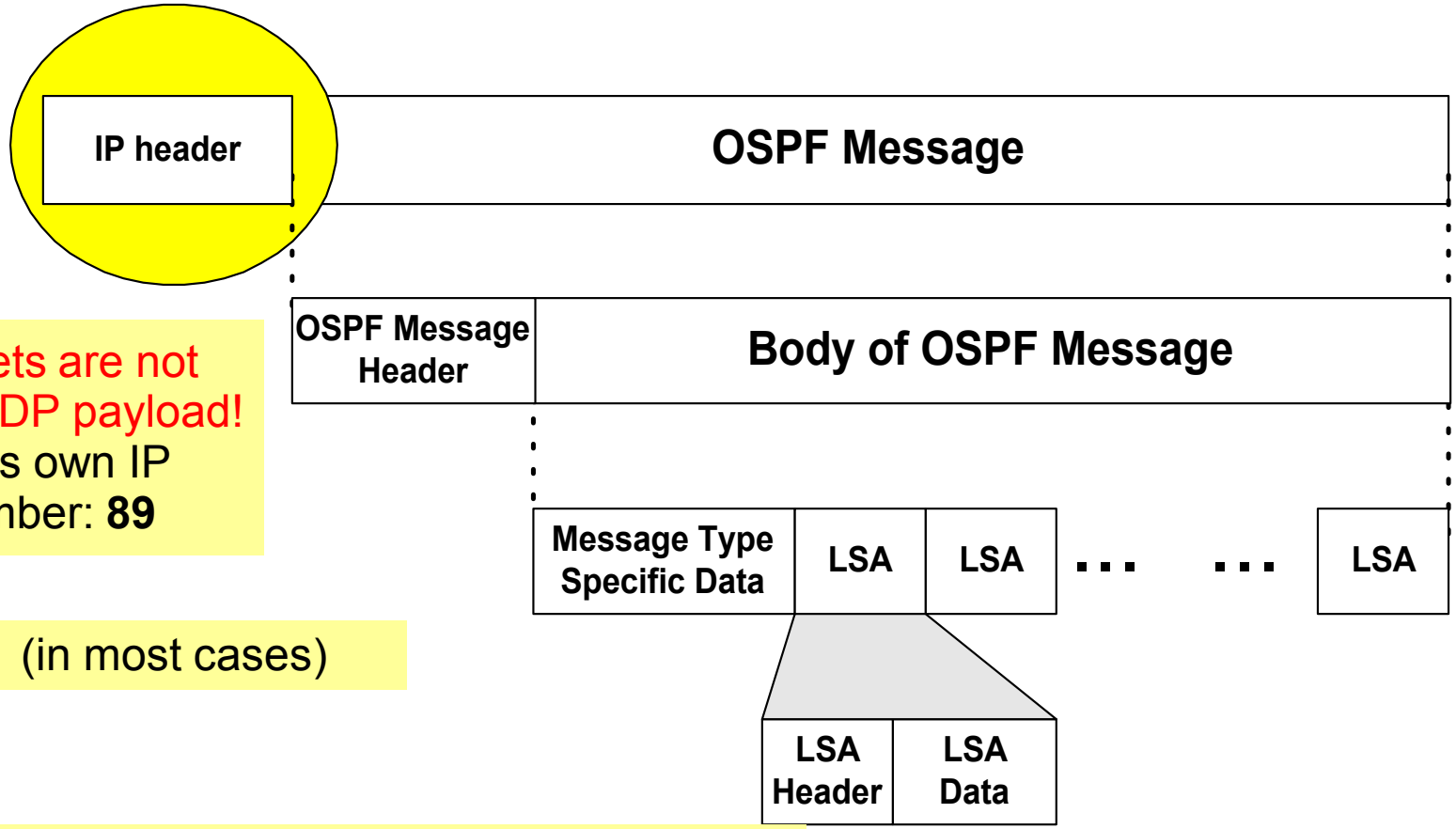


LS Type	Link StateID	Adv. Router	Checksum	LS SeqNo	LS Age
Router-LSA	10.1.10.1	10.1.10.1	0x9b47	0x80000006	0
Router-LSA	10.1.10.2	10.1.10.2	0x219e	0x80000007	1618
Router-LSA	10.1.10.3	10.1.10.3	0x6b53	0x80000003	1712
Router-LSA	10.1.10.4	10.1.10.4	0xe39a	0x8000003a	20
Router-LSA	10.1.10.5	10.1.10.5	0xd2a6	0x80000038	18
Router-LSA	10.1.10.6	10.1.10.6	0x05c3	0x80000005	1680

Link State Database

- The collection of all LSAs is called the link-state database
- Each router has and identical link-state database
  - Useful for debugging: Each router has a complete description of the network
- If neighboring routers discover each other for the first time, they will exchange their link-state databases
- The link-state databases are synchronized using reliable flooding

OSPF Packet Format

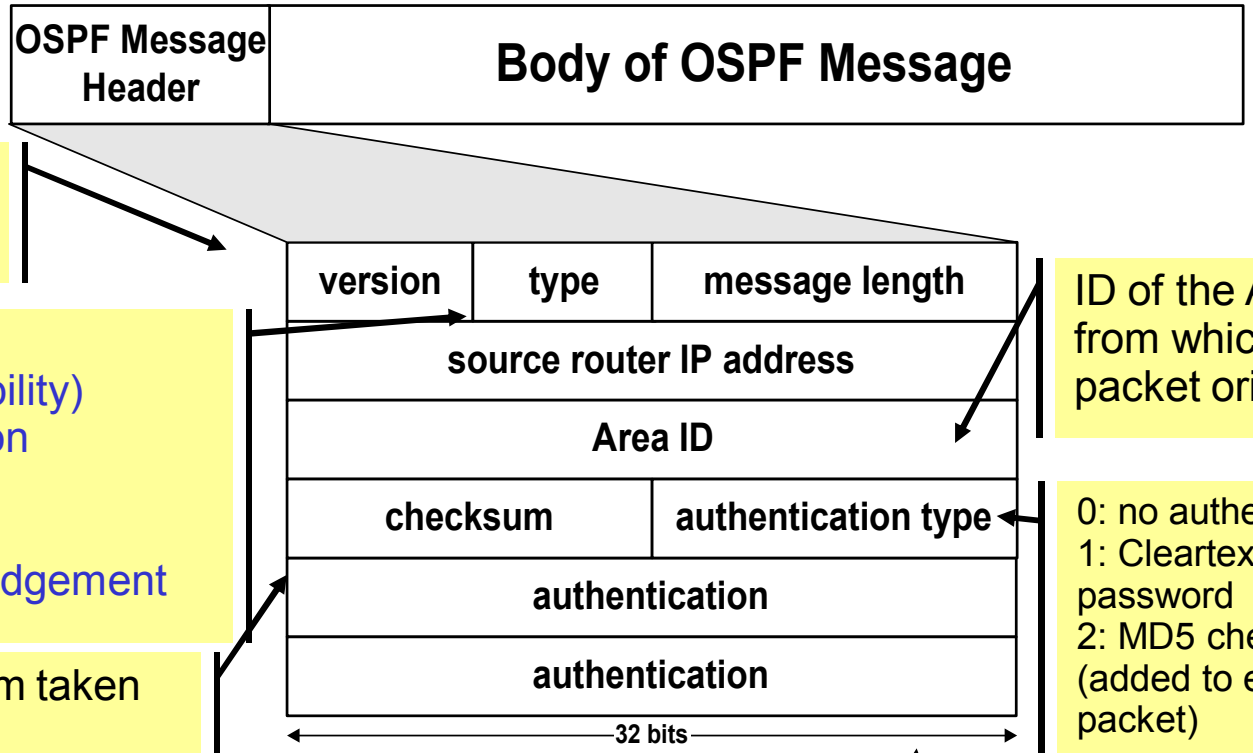


OSPF packets are not carried as UDP payload!
OSPF has its own IP protocol number: **89**

TTL: set to 1 (in most cases)

Destination IP: neighbor's IP address or 224.0.0.5 (ALLSPFRouters) or 224.0.0.6 (AllDRouters)

OSPF Packet Format



2: current version is OSPF V2

Message types:
1: Hello (tests reachability)
2: Database description
3: Link Status request
4: Link state update
5: Link state acknowledgement

Standard IP checksum taken over entire packet

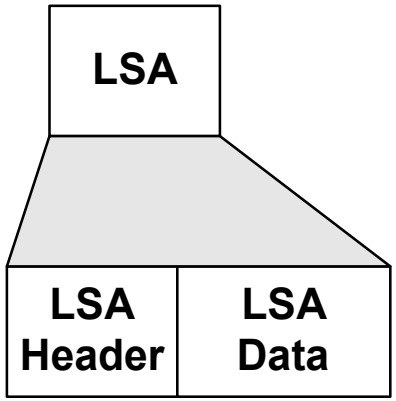
ID of the Area from which the packet originated

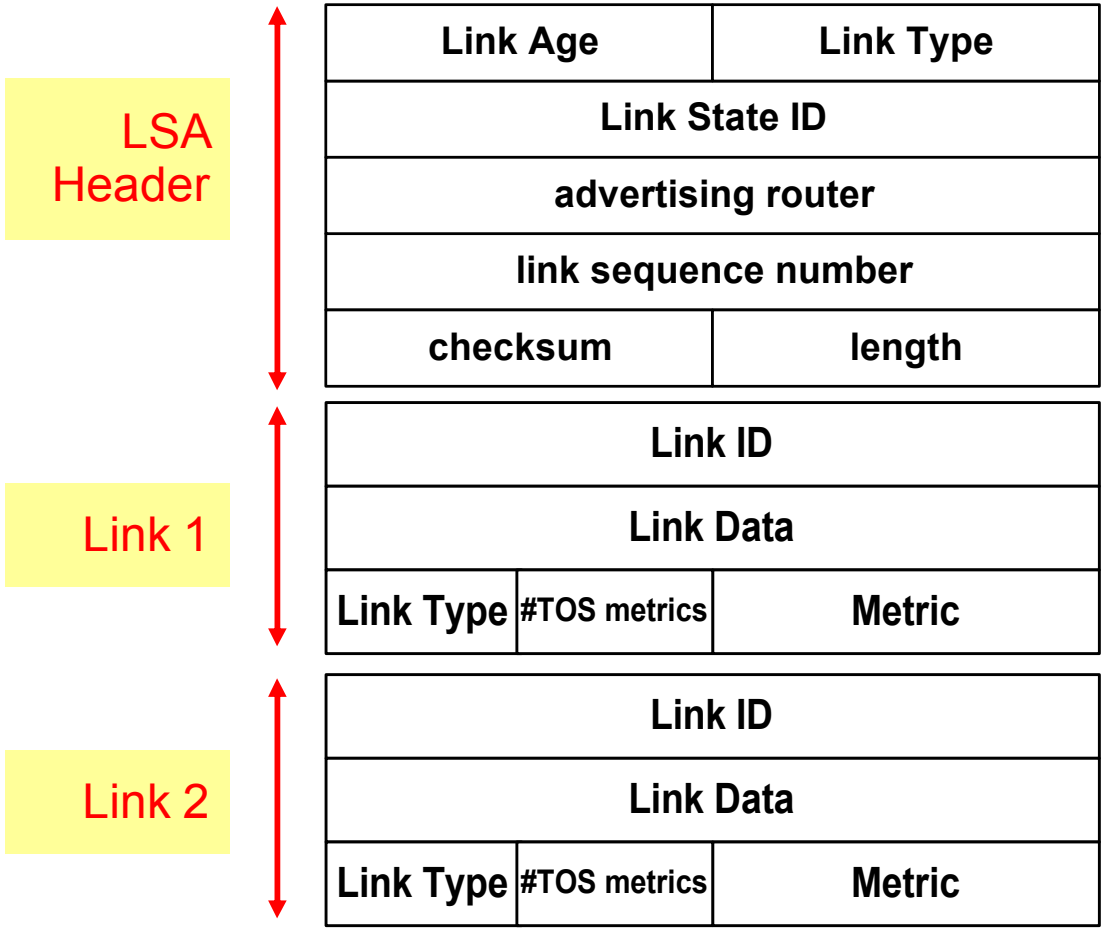
0: no authentication
1: Cleartext password
2: MD5 checksum (added to end packet)

Authentication passwd = 1: 64 cleartext password
Authentication passwd = 2: 0x0000 (16 bits)
KeyID (8 bits)
Length of MD5 checksum (8 bits)
Nondecreasing sequence number (32 bits)

Prevents replay attacks

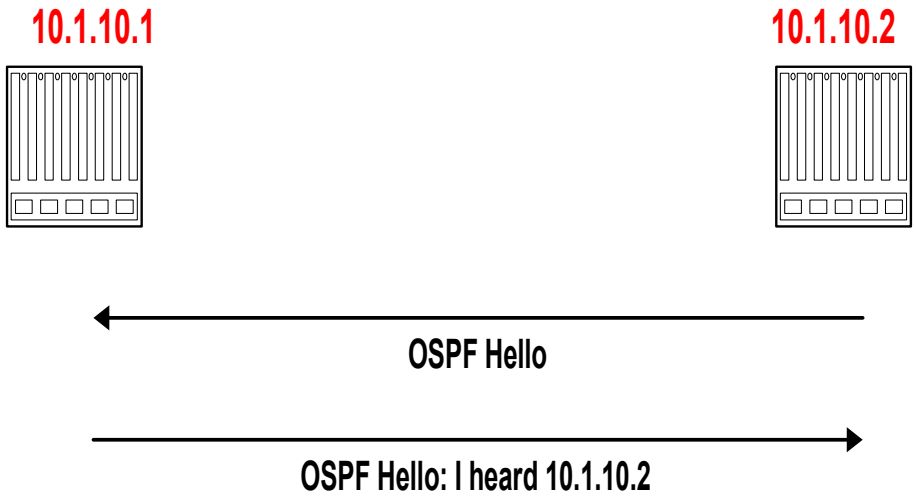
OSPF LSA Format





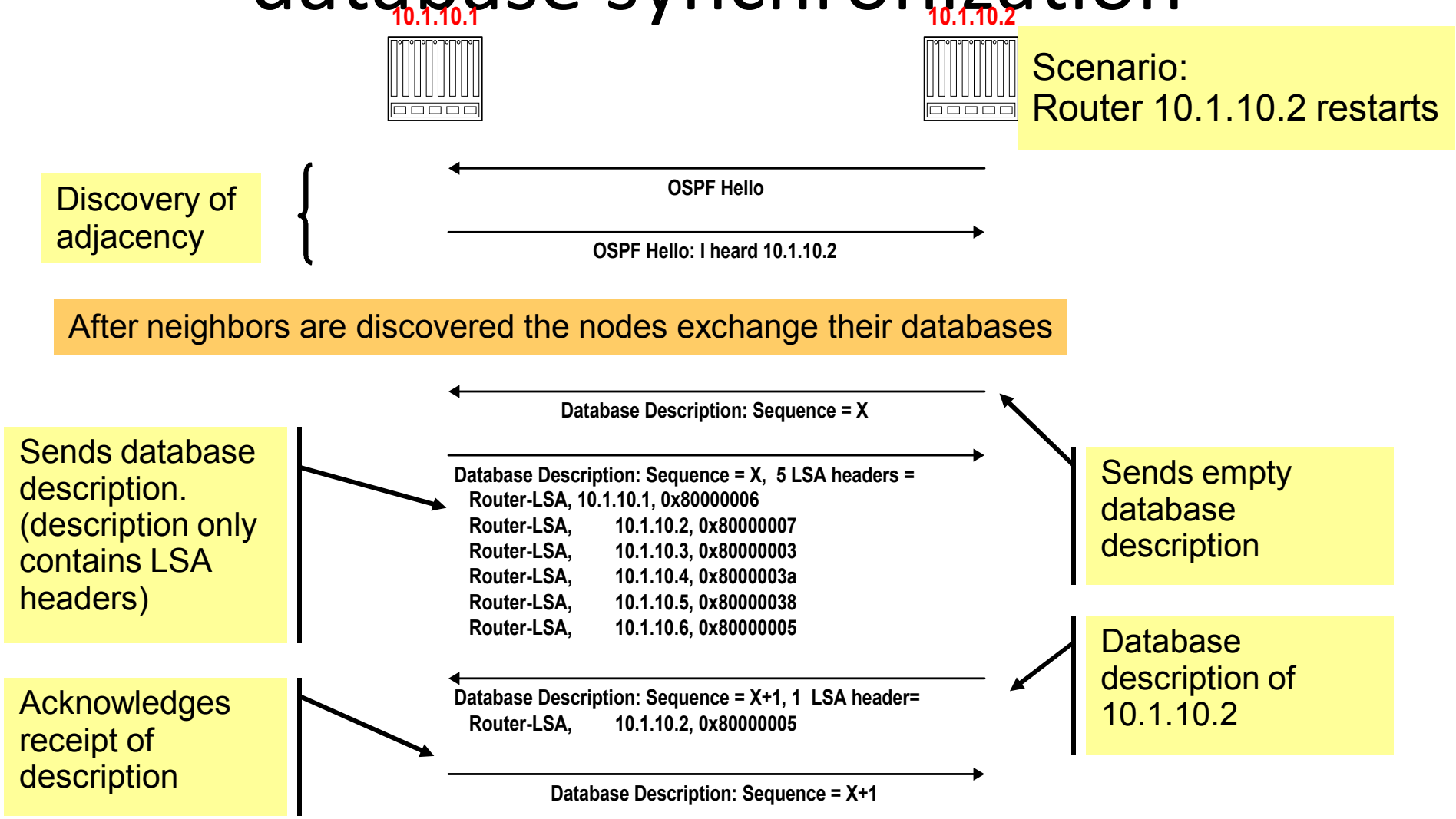
Discovery of Neighbors

- Routers multicasts OSPF Hello packets on all OSPF-enabled interfaces.
- If two routers share a link, they can become neighbors, and establish an adjacency

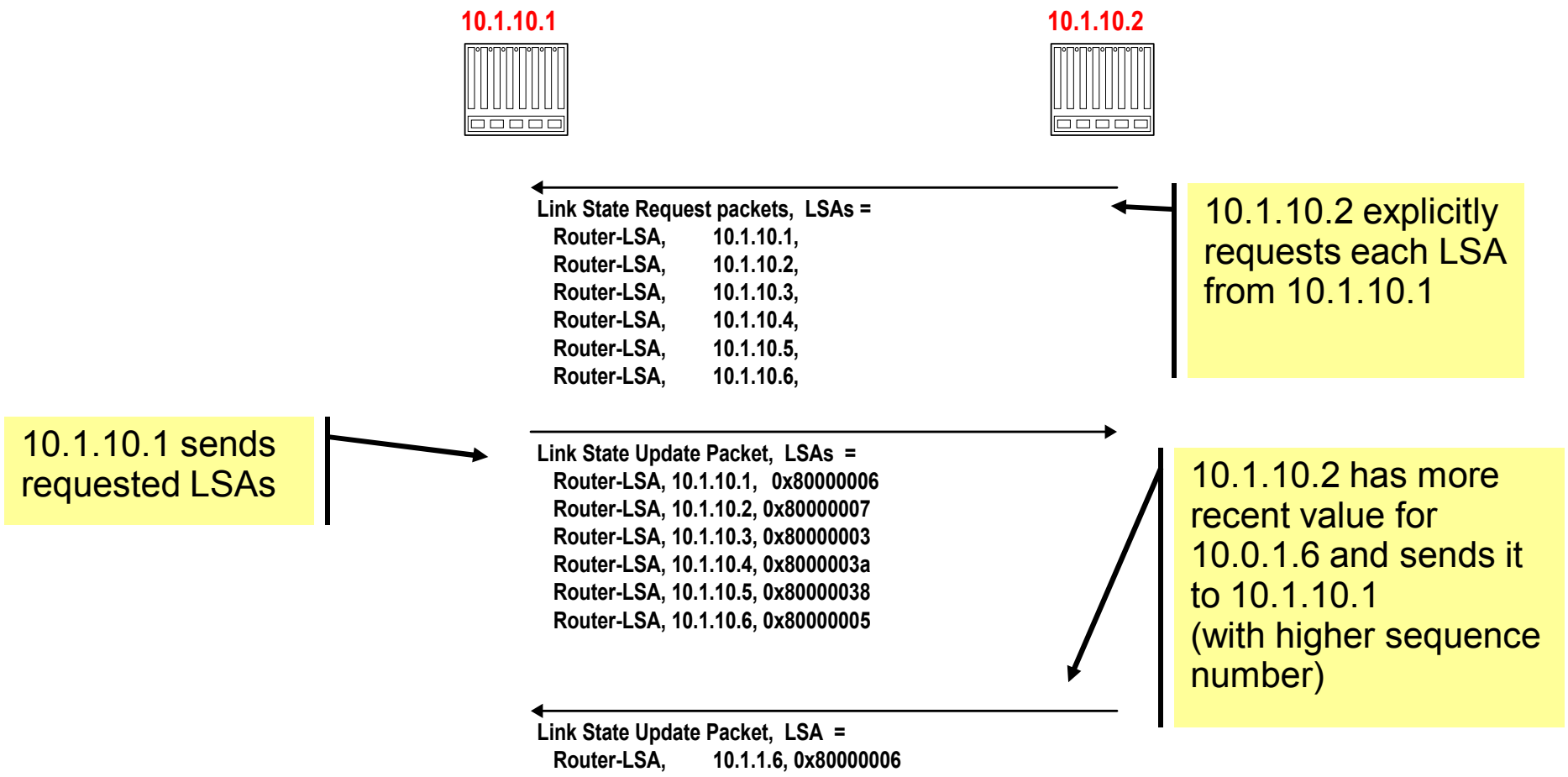


Scenario:
Router 10.1.10.2 restarts

Neighbor discovery and database synchronization

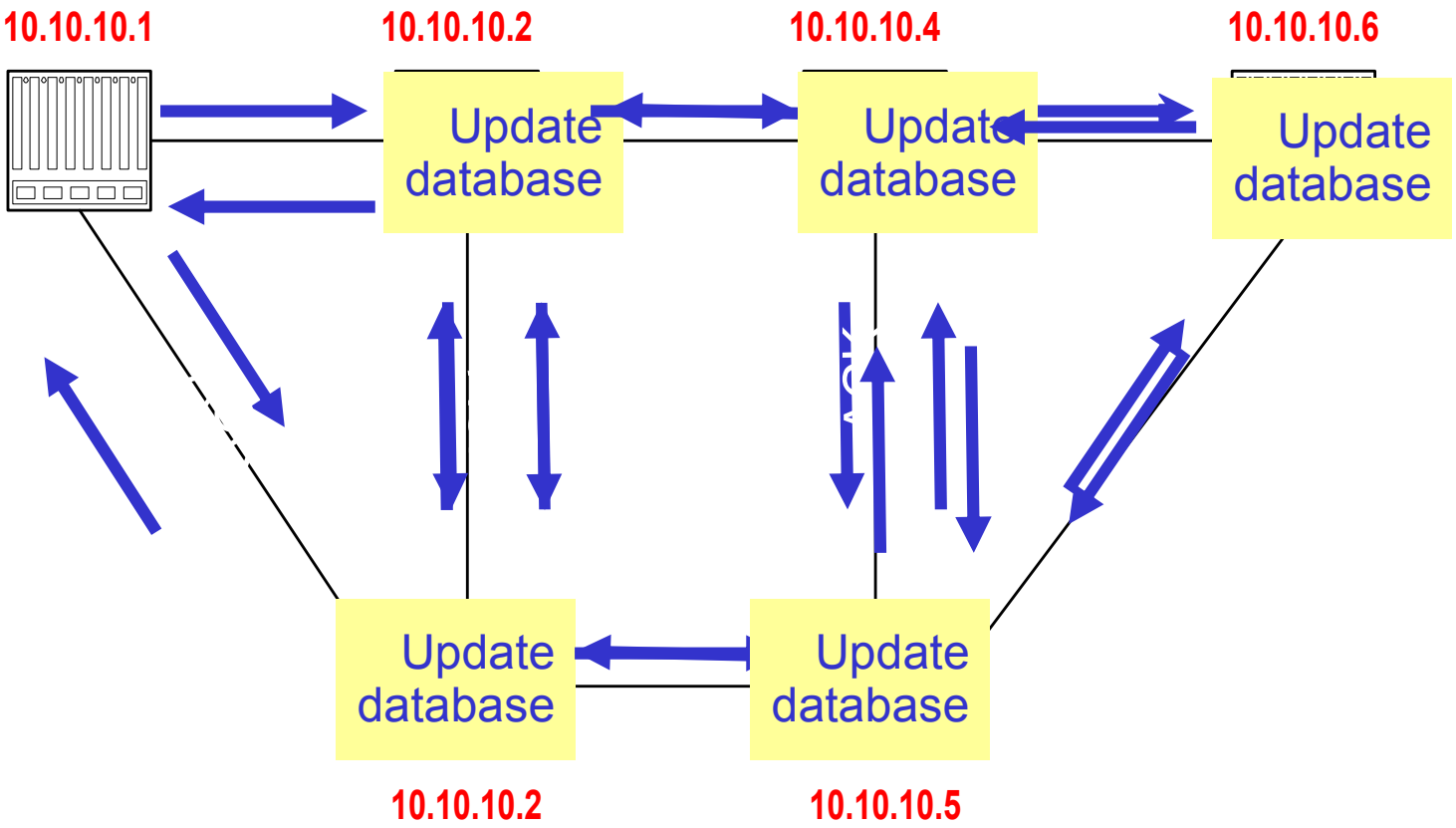


Regular LSA exchanges



Routing Data Distribution

- LSA-Updates are distributed to all other routers via **Reliable Flooding**
- **Example:** Flooding of LSA from 10.10.10.1



Dissemination of LSA-Update

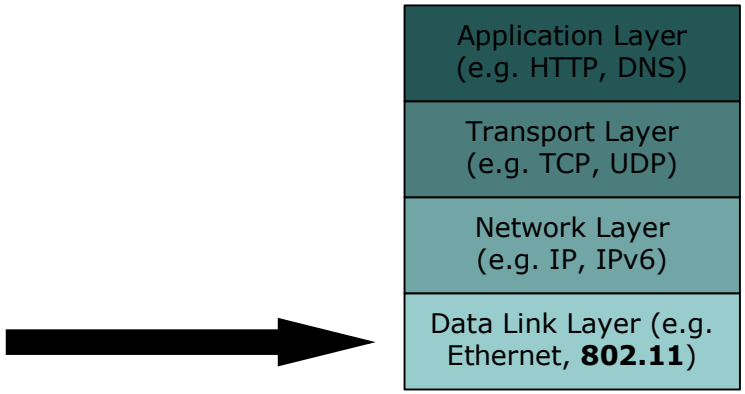
- A router sends and refloods LSA-Updates, whenever the topology or link cost changes. (If a received LSA does not contain new information, the router will not flood the packet)
- Exception: Infrequently (every 30 minutes), a router will flood LSAs even if there are not new changes.
- Acknowledgements of LSA-updates:
    - explicit ACK, or
    - implicit via reception of an LSA-Update

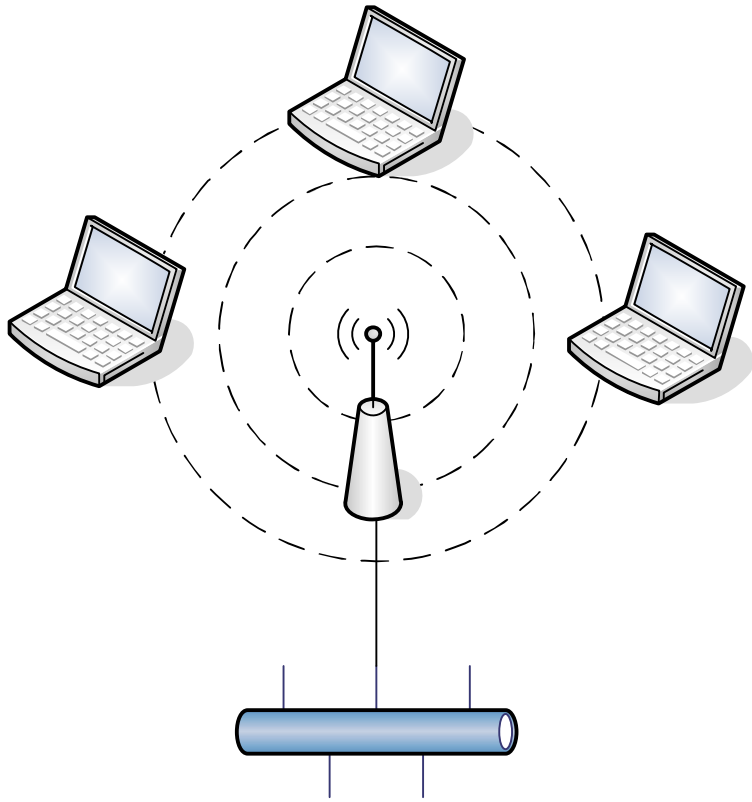
- **Question:** If a new node comes up, it could build the database from regular LSA-Updates (rather than exchange of database description). What role do the database description packets play?

WIFI

Wireless Networking (Wi-Fi)

- Data Link Layer (Layer 2) over radio frequencies
- Many standards
- Notably IEEE 802.11





WiFi's Radio Technology

- WiFi radios that work with the 802.11b and 802.11g standards transmit at 2.4 GHz, while those that comply with the 802.11a standard transmit at 5 GHz.

- Normal walkie-talkies normally operate at 49 MHz. The higher frequency allows higher data rates.

- WiFi radios use much more efficient coding techniques (process of converting 0's and 1's into efficient radio signals) that also contribute to the much higher data rates.

WiFi's Radio Technology (Cont'd)

- The radios used for WiFi have the ability to change frequencies.

- For example, 802.11b cards can transmit directly on any of three bands, or they can split the available radio bandwidth into dozens of channels and **frequency hop** rapidly between them.

- The advantage of frequency hopping is that it is much more immune to interference and can allow dozens of WiFi cards to talk simultaneously without interfering with each other.

802.11b, 802.11a, and 802.11g

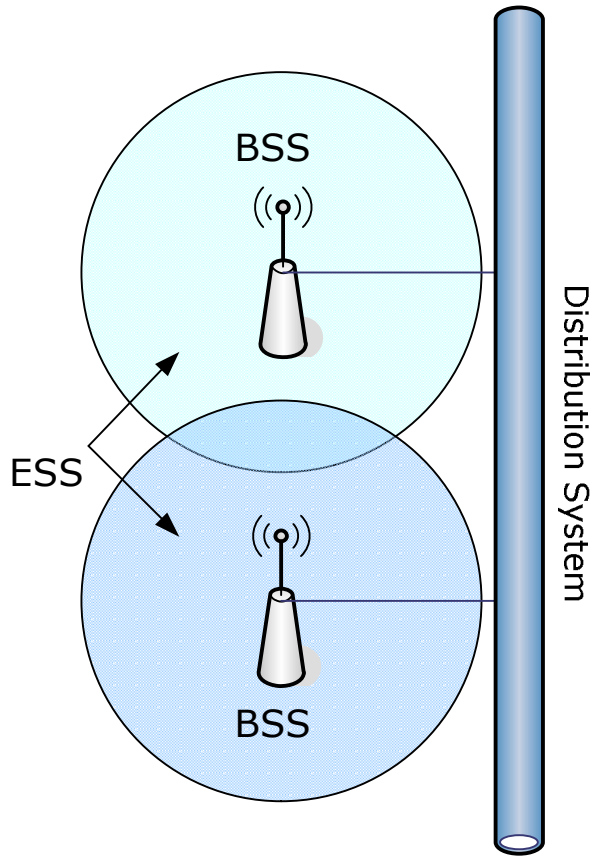
- 802.11b was first to reach the marketplace. It is the slowest and least expensive of the three. 802.11b transmits at 2.4 GHz and go up to 11 Mbps.
- 802.11a was next. It operates at 5 GHz and can handle up to 54 Mbps.
- 802.11g is a mix of both worlds. It operates at 2.4Ghz (giving it the cost advantage of 802.11b) but it has the 54 megabits per second speed of 802.11a. It is also backward compatible to 802.11b.
- Most WiFi cards nowadays are capable of all three of these radio technologies.

802.11b Key Features

- Speeds up to 11Mb/s
  - Scales down to 5.5Mb/s, 2Mb/s, 1Mb/s
  - About half speed taken with overhead
- Uses 13 × 22MHz channels within the IMS (2.4GHz) band in UK
- Omnidirectional range of ~50m
  - Directional, high-gain antennas can transmit over several km
- DSSS, CSMA/CA

802.11 Architecture

- BSS – Base Service Set
  - One cell from one access point
- ESS – Extended Service Set
  - Network of cells
  - Common ESSID
  - Cells linked by DS: Ethernet or wireless (WDS)

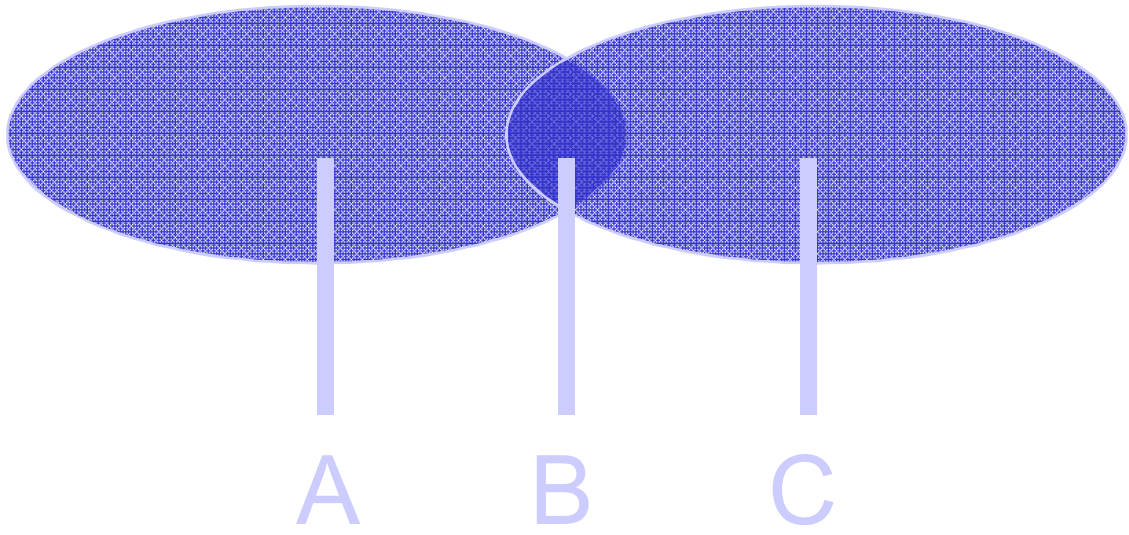


802.11b Layers

- Physical Layer
  - e.g DSSS for 802.11b
- Data Link Layer
  - Media Access Control – CSMA/CA
  - Logical Link Control – 802.2 standard

e.g. HTTP, DNS, SMTP	*Application Layer*
e.g. TCP, UDP	*Transport Layer*
e.g. IP, IPv6	*Network Layer*
802.2 LLC	*Data Link Layer*
802.11 MAC	
DSSS over RF	*Physical Layer*

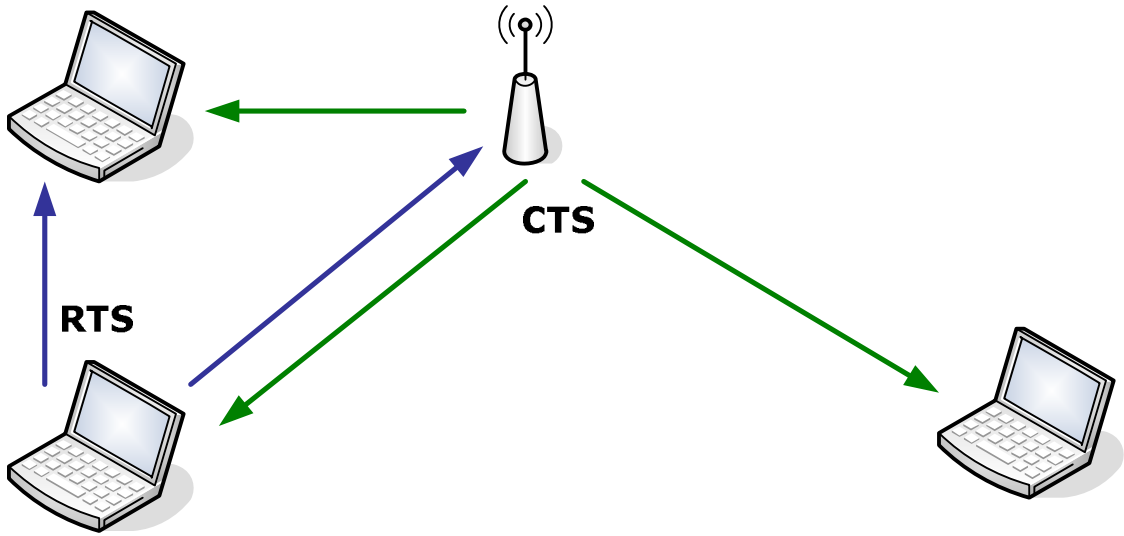
Hidden Node Problem



- A and C cannot see each other, B can see both

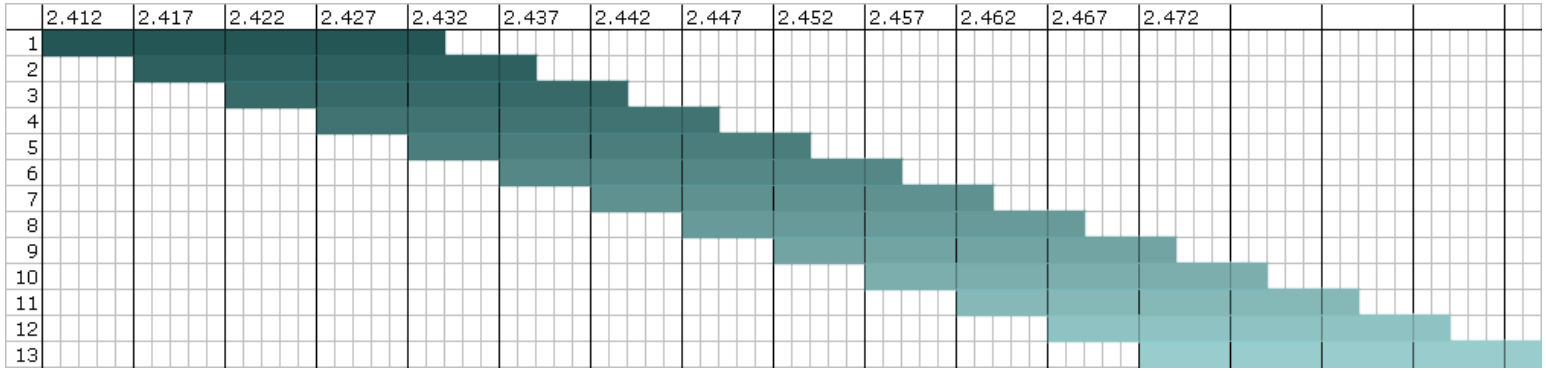
CSMA/CA

- Sender sends *Request to Send* (RTS)
- Receiver sends *Clear to Send* (CTS)
- Sender transmits for required time



802.11b Channels

- In the UK and most of EU: 13 channels, 5MHz apart, 2.412 – 2.472 GHz
- Each channel is 22MHz
- Significant overlap
- Best channels are 1, 6 and 11



TCP Over Wireless

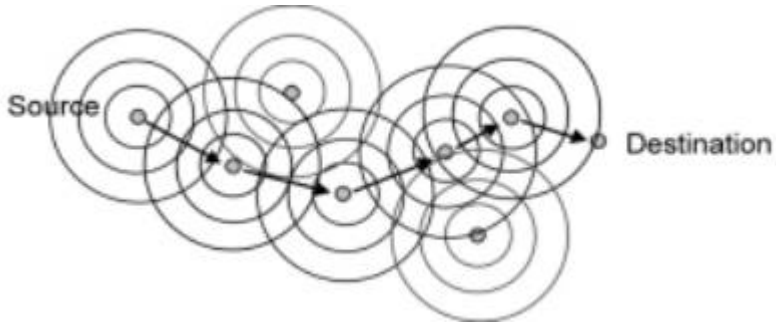
- Wireless unreliable, prone to errors
- TCP will begin a slow start on errors
  - Designed to find optimum window size
  - Inefficient for wireless
- Improvements
  - Adding a threshold
  - TCP Reno – fall back to threshold
- Retransmission Timer
  - Doubles on every retransmission

Security in 802.11b

- WEP
  - *Wired Equivalent Privacy*
  - RC4 and CRC32
  - Known vulnerabilities
- WPA
  - *Wi-fi Protected Access*
  - Larger, dynamically changed keys
- 802.1x
  - Port-based authentication
- 802.11i (WPA2)
  - Builds on WPA
  - AES (Rijndael)

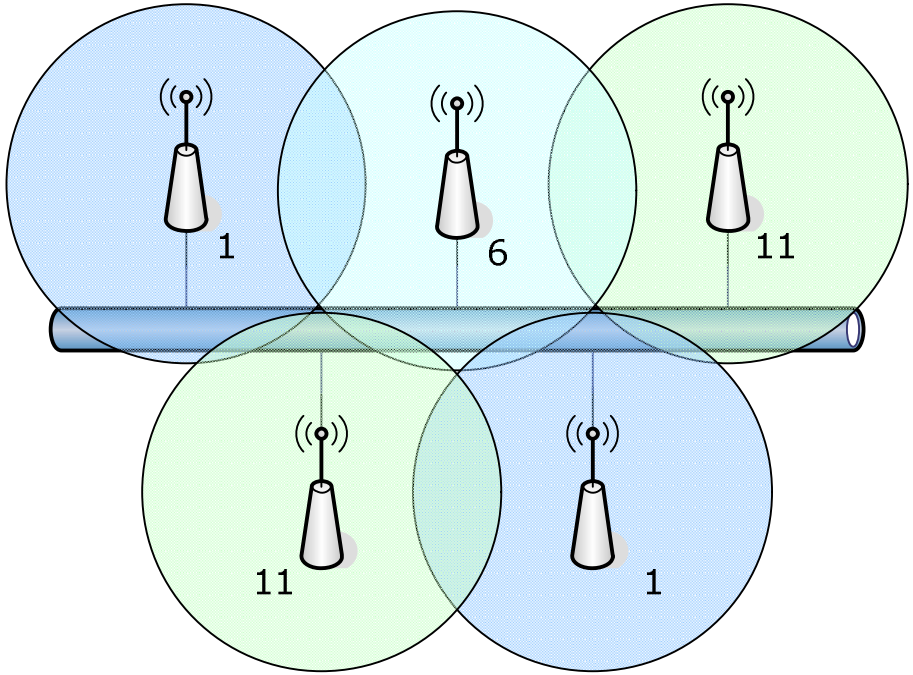
Meshed Networking

- Decentralised infrastructure
- Network of interconnected access points
- Peer-to-peer routing, often redundant



What's *not* a Mesh?

- The ECS Wireless LAN
- Multiple APs
- Different channels
- Same wired subnet

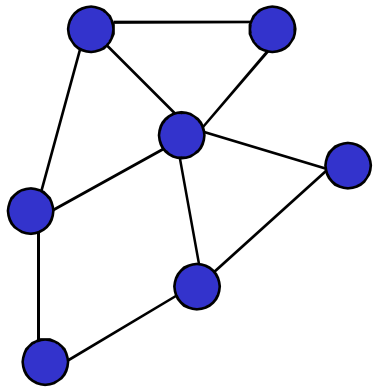


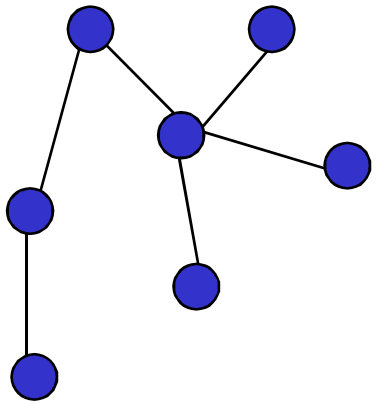
Mesh Approaches

- Ad-Hoc
  - No base station, all hosts are APs
  - Link-local between devices
- Bridging
  - Multiple APs, same subnet
- Routing
  - Multiple APs, multiple subnets
  - WDS links between nodes

Bridging vs Routing

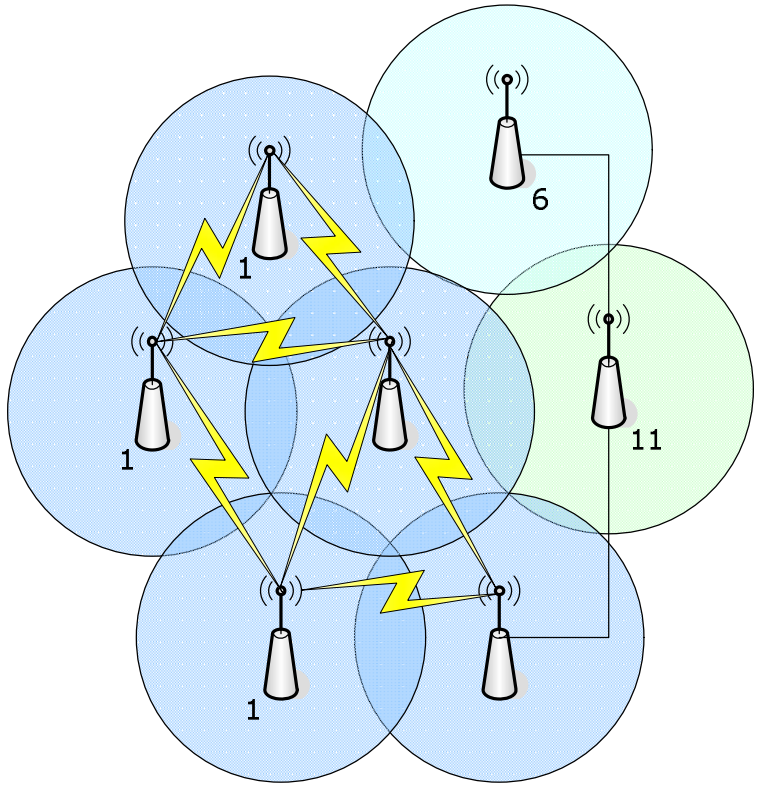
- Bridging gives one large subnet
- Routing permits multiple paths and external links, reduces bottlenecks
- Bridging permits easy mobility





Routed Mesh Networks

- WDS
  - Wireless Distribution System
  - Wireless links between APs, as opposed to wired.
  - All are on same channel.
- OSPF
  - Shortest path routing protocol

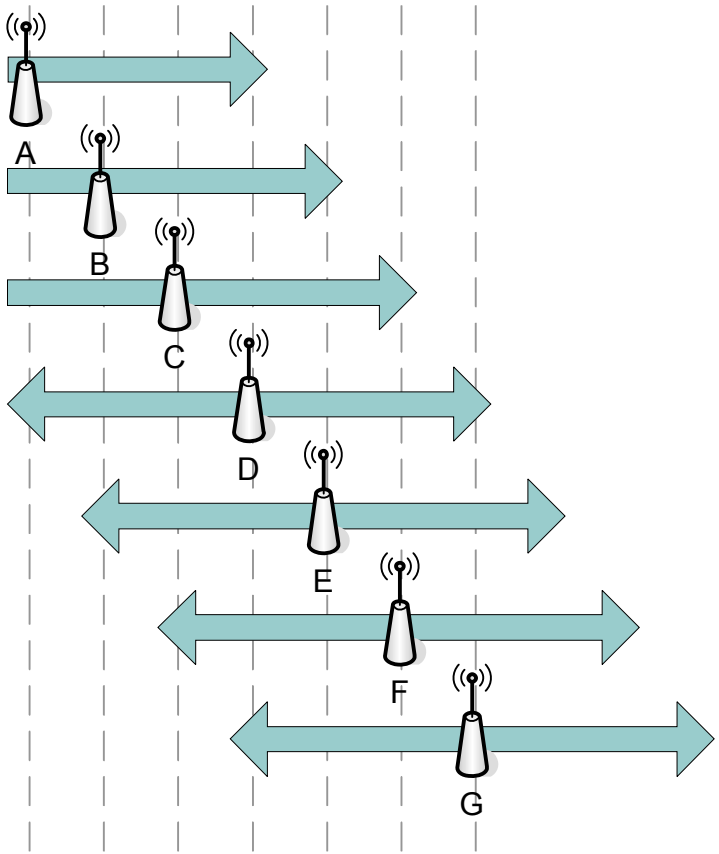


Mesh Issues

- Scale
  - Bridging does not scale well
  - Single-channel WDS does not scale well
- Distances
  - 1km+ distances are possible…
  - …signal degrades, more users
- Congestion
  - Leads to decreasing performance
  - Colliding channels, hidden node

Mesh Throughput Degradation

- All meshed APs on same channel
- Data sent from A to G
- A – B, full throughput
- B – C, ½ throughput
- C – D, ¼ throughput
- D – E, ⅛ throughput
- A now out of range…
- E – F, ⅛ throughput
- F – G, ⅛ throughput



Configuring a Hotspot

- Most wireless access points come with default values built-in.
- Once you plug them in, they start working with these default values.
- However, you may want to change things.
- You normally get to set three things on your access point.

Things to Configure in a Hotspot

- The SSID -- Service Set IDentifier is a sequence of charactersthat uniquely names a WLAN.
  - It will normally default to the manufacturer's name (e.g. "Linksys" or "Netgear").
  - You can set it to any word or phrase you like.
- The channel – the radio link used by access point/router to communicate to wireless devices.
  - Normally it will default to channel 6.
  - However, if a nearby neighbor is also using an access point and it is set to channel 6, there can be interference. Choose any other channel between 1 and 11.

Things to Configure (Cont'd)

- The WEP key -- The default is to disable WEP.
  - If you want to turn it on, you have to enter a WEP key and turn on 128-bit encryption.
  - WEP can be in text format.

Access points come with simple instructions for changing these three values. Normally you do it with a Web browser. Once it is configured properly, you can use your new hotspot to access the Internet from anywhere in your home.